COMPOSITIONAL AGENT-BASED MODELS FOR ELECTRICITY DISTRIBUTION NETWORKS

Fanny Boulaire

Diplôme d'ingénieur/Msc Mathematics and Modelling

A Thesis by Publication submitted in
Fulfilment of the Requirement for the Degree
of
Doctor of Philosophy

Creative Industries Faculty
School of Design
Queensland University of Technology
Queensland, Australia
2015

Keywords

Agent-based models, agent-based modelling

Agent-based modelling and simulation

Complex systems

Composition

Modularity

Network planning

Electricity demand

Renewable energy

Decentralised generators

Socio-technical systems

Abstract

With an increasing number of decentralised generator installations (e.g. rooftop solar panels, batteries, small-scale wind turbines), patterns of electricity consumption are changing over the electricity network. Understanding how electricity flows might evolve over a distribution network as new technologies come into play is important for decision-makers to plan the future grid.

Agent-based modelling and simulation (ABMS) can provide some insight into this matter. This bottom-up modelling technique captures individuals' behaviours and their interactions with one another using simple rules, with the aim to discover emergence at the system level. Such a technique is well suited to model technological change for the electricity distribution network, especially as the past is no predictor of the future, and small changes at a fine geographical and temporal scale might result in large-scale outcomes at the system level. However, because each entity and its relationship to others are described, it can become difficult to keep track of the connections as the system evolves. Consequently, extending and/or modifying the model can become challenging.

This thesis proposes that large-scale agent-based models of networked physical systems be built in a compositional manner to provide extensibility and flexibility in building the models and simulations, with a view to reduce the computational load and improve simulations setup times. MODAM (MODular Agent-based Model) is the software framework that was developed for this purpose. MODAM's software architecture, the models and the simulations are built using modularity and composition. While modularity can be seen as a common feature of ABMS with the software environment following principles from component-based software engineering (with the separation of the simulator, the model, the user interface and the databases), and the agents commonly implemented in distinct classes using the object-oriented paradigm, a central location is still used to set up the agents' relationships to one another, which is done manually. The method proposed in this thesis is novel in that the agents, whose physical and behavioural properties are defined in distinct components, come together in an automated manner, so that an agent-based model and simulation is built at runtime.

The specificities of the modular implementation were driven by the characteristics of the research application tasks with consideration of the tree structure of the distribution network, the integration of new technologies and the massive database requirements used to populate the model, in addition to the fundamental requirements of flexibility and extensibility.

The MODAM framework is an aid to building large-scale ABMS. Its compositional approach not only facilitates building the model incrementally, it also offers many options when setting up simulations, giving flexibility when building the ABMS. As a modeller creates new agent types in independent components, a library of building blocks is being developed. A user can then choose those required for their simulation, without the need to program, which is of benefit for planners who can then analyse scenarios of possible futures by varying the types of agents likely to impact their network. This compositional approach to building ABMS of networked structures is unique.

MODAM was applied to two studies over Townsville, Australia, to understand the impact of technological change on the electricity distribution network. The first application focussed on the impact of photovoltaics on some areas of the network, and how they could be used to reduce the peak electricity demand for different types of users. A more sophisticated study was then developed that illustrates the extensibility and flexibility capabilities of MODAM. Electric vehicle agents were added to the model in addition to the rooftop photovoltaic and consumer agents, and their uptake varied over the years, informed from uptake projection maps. In both cases, the simulations provide details of the load flowing through each asset over the network at half-hourly intervals, capturing its variability in time and space. These outputs can then be displayed using geographical information system applications, so that fast identification of network assets that are at risk of overloading with the introduction of new technologies can be identified. Having such detail of the distribution network assets over which the agents evolve as part of the agent-based model is novel. It provides significant benefits for planners who can make better informed decisions about which assets might need replacing or upgrading when planning their networks.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ABM	Agent-Based Model, Agent-Based Modelling
ABMS	Agent-Based Model and simulation, Agent-Based Modelling and Simulation
ABS	Australian Bureau of Statistics
AC	Alternative Current
AEMO	Australian Energy Market Operator
AER	Australian Energy Regulator
AIC	Akaike Information Criterion
BEV	Battery Electric Vehicle
BOM	Bureau of Meteorology
CCD	Census Collection District
DG	Decentralised Generators
DSM	Demand-side Management
EMF	Eclipse Modeling Framework
EV	Electric Vehicle
FGF	Future Grid Forum
FIPA	Foundation for Intelligent Physical Agents
GHG	Greenhouse Gas
GIS	Geographical Information System
GUI	Graphical User Interface
JADE	Java Agent Development Framework
LRET	Large-scale Renewable Energy Target
LV	Low Voltage

MASON	Multi-Agent Simulator of Neighbourhood/Networks
MODAM	MODular Agent-based Model
M&S	Modelling and Simulation
MV	Medium Voltage
NEM	National Energy Market
NIRAP	National and International Research Alliance Program
NSW	New South Wales
ODD	Overview, Design concepts, and Design Details
OSGi	Open Services Gateway initiative
PEV	Plugin Electric Vehicles
PHEV	Plugin Hybrid Electric Vehicles
PSO	Particle Swarm Optimisation
PV	Photovoltaic
RET	Renewable Energy Target
RCP	Rich Client Platform
SRES	Small-scale Renewable Energy Scheme
SWER	Single Wire Earth Return
TMY	Typical Meteorological Year
UML	Unified Modelling Language
VIF	Variation Inflation Factor
V&V	Verification and Validation

Contributions

- Development of an agent-based model of electrical flows over a large distribution network subject to technological change. This model captures the actual distribution network's physical properties and connectivity, and the actors' behaviours impacting it;
- Development of a novel method that facilitates building large-scale agent-based models and simulations incrementally, as well as offers many options for agents to be defined using alternative, or a combination of, behaviours. It uses modularity and composition and is designed to address the specific features and requirements of the study domain:
  - networked structure,
  - new technologies integration,
  - massive database options,
  - need to analyse different aspects of the behaviour of agents under a range of scenarios;
- Delivery of a software framework, MODAM, that supports the development of scalable, networked structured, agent-based models and simulations, featuring:
  - Ease in extending the model as more information becomes available, which can be done by independent authors without the need to access previously written code,
  - Handling of a wide range of data types and formats, both as input or output to the simulations,
  - High speed of the simulation thanks to a parallel implementation of the scheduler tailored to the networked structure of the ABM;
- Delivery of a library of building blocks for agent-based models, within the electricity distribution domain, that a user can choose from to build simulations;
- Development of an architecture that enables automated agent composition for large-scale agent-based models. A user can choose the agents that are

required for their simulation from the library of building blocks, without the need to program;

- Simulations of technological uptake over a 20-year horizon to understand their impact on grid assets of interest (e.g. transformers). These are used by planners to inform their decisions when planning their distribution grid.

List of Publications

The Queensland University of Technology (QUT) allows the presentation of a thesis for the degree of Doctor of Philosophy in the format of papers, where such papers have been published, accepted or submitted during the period of candidature.

This thesis is composed of six papers, which have been published, and one which was later modified, published and presented at a conference. Note that each paper is selected for this thesis as one chapter, apart from one which is given in the appendix.

These six papers are listed below.

<u>Peer reviewed journal articles</u>

Boulaire, F., Utting, M., & Drogemuller, R. (2015). Dynamic Agent Composition for Large-Scale Agent-based Models. *Complex Adaptive Systems Modeling*. doi: 10.1186/s40294-015-0007-2

Boulaire, F., Utting, M., & Drogemuller, R. (2014). Assessment of the impact of technological uptake on an Australian electricity distribution network through simulations using MODAM. *Environmental Modelling & Software*.

<u>Peer reviewed international conference articles</u>

Boulaire, F., Utting, M., Drogemuller, R., Ledwich, G., & Ziari, I. (2012, 9-12 December 2012). *A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network.* Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.

Boulaire, F., Utting, M., Drogemuller, R., Abeygunawardana, A., Ledwich, G., & Bell, J. (2012). *Planning for the Impact of Distributed Solar Energy on the Grid*. Paper presented at the Solar 2012 Conference, Swinburne University of Technology, Melbourne.

Boulaire, F., Utting, M., & Drogemuller, R. (2013b, 26/08/2013). *Parallel ABM for electricity distribution grids: a case study*. Paper presented at the 1st Workshop on Parallel and Distributed Agent-Based Simulations, Euro-Par 2013, Aachen, Germany.

Boulaire, F., Utting, M., & Drogemuller, R. (2013a, 18-26 May 2013). *MODAM: A MODular Agent-based Modelling Framework.* Paper presented at the 2nd International Workshop on Software Engineering Challenges for the Smart Grid as part of 35th International Conference on Software Engineering (ICSE 2013), San Fransisco, CA, USA.

Additionally, the following items are part of the work undertaken during this PhD but do not constitute chapters in this thesis.

<u>Peer reviewed journal article</u>

Boulaire, F., Higgins, A., Foliente, G., & McNamara, C. (2013). *Statistical modelling of district-level residential electricity use in NSW, Australia*. Sustainability Science, 1-12. doi: 10.1007/s11625-013-0206-8

<u>Peer reviewed international conference article</u>

Utting, M., & **Boulaire, F.** (2015). *Specification and Validation of the MODAM Module Manager*. Paper presented at the 2nd International Workshop about Sets and Tools (SETS 2015) as part of the 20th International Symposium on Formal Methods, Oslo, Norway.

Drogemuller, R, **Boulaire, F.**, Ledwich, G., Buys, L., & Utting, M., Vine, D., Morris, P., Arefi, A. (2014). *Aggregating energy supply and demand*. Paper presented at the 10th European Conference on Product & Process Modelling - ECPPM 2014, Vienna, Austria.

<u>Items published and presented at conferences</u>

Boulaire, F., Utting, M., & Drogemuller, R. (2013). *Modelling for the Electricity Distribution Network.* Extended abstract presented at the MODSIM 2013, Adelaide, South Australia.

Boulaire, F. (2013). Building Adaptable Agent-Based Models – Application to the Electricity Distribution Network. Extended abstract presented at the MODSIM 2013, Adelaide, South Australia.

Scholarships and Awards

- Recipient of the Wal Read Award 2012 – Best Postgraduate Paper for:

  Boulaire, F., Utting, M., Drogemuller, R., Abeygunawardana, A., Ledwich, G., & Bell, J. (2012). *Planning for the Impact of Distributed Solar Energy on the Grid*. Paper presented at the Solar 2012 Conference, Swinburne University of Technology, Melbourne.

- Grant from SIGSIM to attend the 2012 Winter Simulation Conference in Berlin, Germany

- Grant from SIGSOFT to attend the 2013 ICSE conference in San Francisco, USA.

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

QUT Verified Signature

Signature:

Date: 05/11/2015

Acknowledgements

First of all, I would like to thank my main supervisor Professor Robin Drogemuller for giving me this opportunity of doing a PhD, and for his mentorship over the many years we have worked together, especially the last three. I also would like to acknowledge with sincere gratitude my associate supervisors Professor Gerard Ledwich, for his patience as I was getting familiar with the world of electrical engineering, and Dr Andrew Higgins for his guidance and insights into the world of research. Also, a big ‘Thank You’ to Dr Mark Utting, my colleague, who shared his wealth of knowledge with me and was of great support.

This PhD journey might have lasted three years only, but it is the outcome of many years of learning, which desire has been nurtured by my parents. Without their love, sacrifices and encouragements, this journey might have never eventuated. To them, I am immensely grateful.

To the rest of my family: brothers, sister, Australian adoptive parents and friends, whether they are far away or close by. For their relentless questions concerning my thesis subject, and their care, laughter and love when my answers did not make sense.

Finally, to the One, from whom all good things come. This PhD journey, which started has a blind step in faith, has been an immense blessing to my life. Meeting Thuy, now my husband, was an unthinkable blessing and has given this PhD journey a different light, a much brighter one. And now, with this new life growing in me, a new path opens up. May we all continue growing in knowledge and faith, however small it is, so that we can in turn be a blessing to others.

Chapter 1: Introduction

The research presented in this thesis presents a novel approach to building flexible and extensible large-scale agent-based modelling and simulation applications (ABMS applications) where the system represented is the electricity distribution grid. This chapter gives the background to this research in the first instance, followed by the research problem and the aims of the study. Finally, an account of the research progress linking the research papers, presented as chapters in this thesis, is given.

The focus of this research is in supporting complex systems modelling.

1.1 BACKGROUND AND MOTIVATION

Australia has seen a steady increase in electricity consumption over the last few decades, with an average increase of 2.8% a year over 2000-2010 in Australia (Cuevas-Cubria, Schultz, Petchey, Maliyasena, & Sandu, 2010). While consumption was expected to continue on increasing, this has not been the case since 2010, which has led AEMO (Australian Energy Market Operator) to readjust their predictions down in 2013, not only for the energy but also for peak demand (Australian Energy Market Operator, 2013) across the states of South Australia, Victoria, Tasmania, New South Wales and Queensland. The reasons identified by AEMO for this decrease are "*continued increases in rooftop photovoltaic systems and energy efficiency savings from new building regulations [which] have offset growth in residential, commercial and light industrial annual energy*" (Australian Energy Market Operator, 2013). In addition to these two factors, changes in the economy away from electricity intensive industries (shutdown of some industries), and a response of electricity consumers to higher electricity prices (since 2010) was also suggested by (Saddler, 2013) as contributing to the overall decrease. Despite this recent observed decrease, the adjusted 10-year outlook (2013–14 to 2022–23) drawn by AEMO still sees the annual energy forecast to grow by 1.3%, explained by population growth projections, however lowered from the initial 2.4% prediction (Australian Energy Market Operator, 2013).

While in the past few decades, the annual average consumption had been growing approximately in proportion with population growth, it is not the case anymore as highlighted above. Other factors which are influencing the consumption are starting to have more of an impact than they did in the past, leading to a shift in paradigm, where solely studying the past is no longer sufficient to predict the future. This is further exacerbated with new technologies that are expected to come on the market, but for which we have very little insight in to how they might affect the network. This is the case for battery storage and electric vehicles, which in addition to renewable energy systems have been identified by (Manyika et al., 2013) amongst the 12 disruptive technologies that could have a big impact on the economic and societal landscape by 2025.

While labelled disruptive, these technologies can however have a positive outcome if their characteristics are properly understood and their potential harnessed. After all, these technologies, such as the renewable energy ones, are the result of conscious decisions. Indeed, following commitments made to ensure that Australia limits its pollution in the future, measures have been taken by the Australian Government: the Kyoto protocol was ratified by Australia in 2007 (Australian Government, 2010) and further commitments, such as reaching 20% of renewable energy generation by 2020 (Australian Government, 2011a), were made to cut pollution by 80% below the 2000 levels by 2050 (Australian Government, 2011a). To increase the share of the electricity generation from renewable sources, the Renewable Energy Target (RET) scheme (Australian Government, 2011b) was introduced. It is made of two parts, the Large-scale Renewable Energy Target (LRET) (Australian Government - Clean Energy Regulator, 2012a) that is applied to generation at the high-voltage level with concentrated solar power systems for example, and the Small-scale Renewable Energy Scheme (SRES) (Australian Government - Clean Energy Regulator, 2012b) which encourages individuals to invest in small-scale generation units, such as rooftop solar panels (Queensland Government - Office of Clean Energy, 2011). While the current Commonwealth government is intending to remove the RET, the goal of CO_2 emission reduction is still in place. These decentralized generators (DGs) have advantages not only in terms of CO_2 emission reductions, but they may also reduce the impact of peak load growth if installed at strategic locations on the network, potentially saving money on infrastructure investment.

Despite the very valuable introduction of renewable energy technologies in terms of carbon emission reduction and peak shaving potential, new challenges (described below) have emerged for the different electricity providers preventing the installation of renewable DGs in some areas that have reached a high percentage of penetration (Hall, 2011). These types of problems need to be understood if the SRES target of 4,000GWh generation is to be reached in Australia, when in 2011 energy produced by photovoltaic (PV) was estimated at 1,200GWh (AEMO, 2012). Some countries in Europe have managed to reach quite high percentages of variable renewable generation (e.g. Germany peaked at over 50% of its electricity demand in June 2014 (Burger, 2014)), and some models are claiming that an economy based on 100% renewables is possible (Lund & Mathiesen, 2009). Learning from their experiences would be highly beneficial, however, their high rate of renewable penetration can be achieved thanks to cross-border transmission links (Kirby & Milligan, 2008) and strong, dense distribution networks. Australia, being isolated from other countries and having sparse and low density networks, needs to deal with the technical challenges in a different manner.

Examples of the technical challenges that distribution companies are facing are the management of intermittent load flows or network imbalances. Due to weather variability, some of the renewable energy technologies have intermittent power generation over both short and long time frames, as in the case of wind and solar generation. Another challenge with the installation of distributed and dispersed sources of electricity for the low-voltage network lies in the fact that electricity is not only generated, but also consumed, on site. Having both consumption and generation at different points in the network means that bi-directional power flows have to be adequately managed. It has the potential to reduce the need for line capacity increase, but because of their intermittent generation, additional measures have to be taken to ensure continuous electricity delivery to the customer. This can be done either by accessing electricity from the grid (consequently still requiring network management) or from energy storage, which can be charged during times of lesser demand. One form of energy storage that has become popular over the last few years and trialled to support distribution networks is battery storage. In some cases, large batteries have been installed at specific locations on the network (McArdle, 2013) while other companies have offered plans for their customers to have both PV and battery at their home (Parkinson, 2013). In addition to grid usage, it is expected that an increasing number of individuals will take up these technologies for their own benefit, as batteries become more viable. As battery prices decrease and electricity grid prices are set to increase, using them in partnership with PV installations offers great potential. Depending on how these are being used, they might also be of great benefit for the distribution companies if the users are rewarded for helping reduce the peak, which would in turn prevent or delay upgrade or augmentation of the network.

Overall, electricity providers are faced with assessing a range of network options so that the different challenges of meeting fluctuating demand (average and peak), as well as variable and new demand patterns caused by new technologies can be met while improving ageing networks and meeting policy targets with the wider introduction of renewable energy sources and other decentralised technologies. Because of the lack of experience with high penetration rates of renewable electricity generators worldwide, very little is known about what will happen at the larger scales when these generation types increase to higher proportions (Komor, 2009), and even less is known about the impact new technologies taken up by individuals will have when they come on line. Consequently, distribution companies are looking at understanding such effects using modelling techniques that can investigate different scenarios for planning future energy grids.

One modelling technique that has recently gained popularity in the electricity sector is agent-based modelling (ABM) (Batten & Grozev, 2006; Boait, Ardestani, Mark Rylatt, & Richard Snape, 2013; Chappin & Dijkema, 2010; North et al., 2002; Weidlich, 2008). Like other modelling techniques, ABM is used to answer specific questions from real world systems that could otherwise be expensive or impractical. Its recent gain in popularity can be attributed to its capacity to use information at a fine level of detail of the system, both geographically and temporally, and generate information at a higher level where emergent patterns can be observed (House-Peters & Chang, 2011). With new technologies being introduced to the market, and no data on which to base analyses to predict what might happen on the electricity network, using such a bottom-up technique where simple rules define the actions and interactions of the network entities is beneficial, as it can capture in a simple way how they interact, and through their evolution, how the network will be impacted overall, as well as locally. This technique is data-intensive, as explicit data at a fine level of detail is used, and it is computer-intensive as many interactions between agents are required. With the growing availability of data and the increase in computer power, these concerns are however fading. Nonetheless, being able to update or extend an agent-based model as more information becomes available can become problematic, due to the tight coupling of the agents and their dependence on the data, especially when modelling very large systems. This thesis proposes large-scale ABMS applications be built in a compositional manner to answer these challenges.

1.2 RESEARCH AIMS AND QUESTIONS

The aims of this study are

- To build agent-based models so that flexibility and extensibility in building and running the models can be achieved. It is proposed in this thesis that such properties can be achieved using a compositional approach. By composition, we mean that the information describing the model can be broken down into small pieces and held in different components that can be developed and used independently. The information contained in these components can then be connected to one another through a mechanism to form the model on which simulations are run. Such an approach can be seen as a Lego-type of approach, where the individual components can be combined to form very sophisticated models of systems;
- To assess the impact of different trajectories of electricity consumption at key locations of the electricity distribution network over many years. This is done through the development of simulations using scenarios of technological change. These simulations are also used to validate the proposed method of developing the agent-based models.

This leads to the formulation of the research question:

"Can a compositional approach to building large-scale agent-based models support modelling the effects of technological change on an electrical distribution network?"

This thesis has:

- a methodological focus which extends agent-based modelling frameworks proposing a compositional approach to model large complex systems, and
- an application aspect which aims at quantifying electricity consumption including modelling technological change, at fine geographical and temporal scales, and assessing their impact on an electricity distribution network. This application aspect is

used to validate and investigate the effects of composition on large agent-based models.

Three additional sub-questions relating to the compositional method of building large-scale agent-based models can further be identified:

- *How to implement ABMS applications so that new elements can easily be added to an existing model (extensibility), as well as new behaviours (flexibility)?*
- *What improvement in speed and ease of modelling does this approach provide and how to easily build large-scale agent-based models and simulations, without the need to code?*
- *What impact does using a compositional method have on the execution speed of the model and how to improve it?*

These three questions are important because with a growing system, it can become challenging to extend agent-based models with additional agents and maintain the different dependencies amongst them. Providing a mechanism for easily adding information about new elements as new knowledge becomes available will support extensibility of the model – e.g. it is possible to describe new technological systems, such as new types of storage, as they are made available on the market. Also, providing a mechanism so that some elements can be easily added or removed from the model will support flexibility in the development of scenarios – e.g. it is possible to assess the impact a technology might have by comparing the impact of its introduction with a base scenario, as well as assessing the way it is being utilised. Finally, as these mechanisms for building the agent-based model in a flexible and extensible manner are developed, they also need to bring ease and speed in implementing the models, building the scenarios, and running the simulations.

Such an approach is chosen to support the development of simulations where the load, either being consumed (e.g. at a premise) or flowing through each asset (e.g. feeder) over a distribution network, is to be estimated subject to different conditions. This includes simulations of the current condition of the network, as well as simulations using scenarios of possible futures, where decentralised systems might be put in place by individuals, and impact demand over the network differently. Additional sub-questions relating to the application of compositional agent-based models to the electricity distribution network can be described as:

- *How can the agent-based model represent the physical connectivity of the network and integrate it with the behaviour of the individuals using it?*
- *What type of agents can be built? What types of simulations can be run?*
- *How can technological change within the electricity network be modelled and assessed?*

1.3 SPECIFIC OBJECTIVES OF THE STUDY

In order to achieve the mentioned aim, the specific objectives of this study are as follows:

1. Develop a spatio-temporal framework for the integrated assessment of the electricity distribution network
    - Define the boundaries of the dynamic system under study;
    - Develop a framework that supports the examination of the system as a whole, so that the impact of a change in one part of a system on other parts can be assessed;
    - Bring together in an agent-based model the physical structure of the network and the behaviour of the agents.
2. Develop the compositional architecture for the development of agent-based models of physical networks
    - From the limitations of a monolithic approach to building large-scale agent-based models define the requirements for a compositional approach;
    - Implement the architecture of the compositional agent-based model and simulation application ensuring that software and model are built in a compositional manner;
    - Develop a parallel implementation of the scheduler to speed up simulations;
    - Measure the flexibility, extensibility and speed of execution of the compositional approach to building agent-based models.

3. Validate the compositional agent-based modelling framework using simulations to assess the impact of technological change on the electricity distribution network
    - Expand the ABMS application with the addition of new agents or localised adaptation of existing ones;
    - Create simulations of possible futures by mixing and matching agents;
    - Assess the impact of technology uptake from the analysis of simulation outputs.

1.4 ACCOUNT OF RESEARCH PROGRESS LINKING THE RESEARCH PAPERS

This thesis is organised in nine chapters, five of which are papers that have been published in journals or conference proceedings, and one that was later modified and published in a conference proceeding. A summary of how these papers are contributing to the research question is given in this section. Table 1-1 gives an overview of how these papers relate to the objectives of the study, along with the questions these papers investigate and the results obtained.

Objectives of the study	Questions	Results	Chapters written as Papers
(1) Develop a spatio-temporal framework for the integrated assessment of the electricity distribution network	How can electricity consumption be modelled to understand its evolution over time and space with: - current network characteristics - decentralised generation contributing to the network How can the agent-based model represent the physical connectivity of the network and integrate it with the behaviour of the individuals using it?	Use of a combination of modelling techniques: - ABM to represent the system units accurately and dynamically - PSO to find the most economical configuration of DGs	**Chapter 3** A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network
(2) Develop the compositional architecture for the development of agent-based models of physical networks	How to implement the ABMS application so that new elements can easily be added to an existing model (extensibility), as well as new behaviours (flexibility)?	Compositional implementation for the software, the model and the simulation setup where the agents are defined as an asset, one or many behaviours and optional data	**Chapter 4** Dynamic Agent Composition for Large-Scale Agent-based Models This paper was extended from a workshop paper provided in the appendix: "MODAM: A MODular Agent-based Modelling Framework"
	What improvement in speed and ease of modelling does this approach provide and how to easily build large-scale agent-based models and simulations, without the need to code?	Automation of the integration of the components to build the ABMS	**Chapter 5** Large-scale agent-based modelling and simulation - Automation based on a dynamic agent composition
	What impact does using a compositional method have on the execution speed of the model and how to improve it?	Use of the underlying structure of the network to run simulations in parallel	**Chapter 6** Parallel ABM for Electricity Distribution Grids: a Case Study

(3) Validate the compositional agent-based modelling framework using simulations to assess the impact of technological change on the electricity distribution network	What type of agents can be built? What types of simulations can be run?	Development of a weather-driven PV output algorithm	**Chapter 7** Planning for the Impact of Distributed Solar Energy on the Grid
	How can technological change within the electricity network be modelled and assessed? How does the electricity consumption vary over space and time under different scenarios of technological uptake (rooftop solar panels, electric vehicles)?	Mix-and-match of modules to compose simulations of agent-based models. Localised adaptation of behaviours within modules	**Chapter 8** Impact of technology uptake on an Australian electricity distribution network

Table 1-1 - How the papers forming chapters in this thesis contribute to the objectives of the study

Chapter 3 presents a planning tool that was built to support the identification of investment strategies for distribution networks over large areas and long planning horizons. This tool was described in (Boulaire, Utting, Drogemuller, Ledwich, & Ziari, 2012) and was presented at the 2012 Winter Simulation Conference, in Berlin.

This paper presented the two modelling techniques used in the planning tool: agent-based modelling (ABM) and particle swarm optimization (PSO). The PSO was developed by Dr Iman Ziari, as complementary work to the ABM developed by this author. ABM is used to represent the different system units accurately and dynamically, following the changes over time and at different levels of detail in the distribution network. Load duration curves which are output from the ABM simulation are then used in the PSO module to find the most economical mix of network extension and integration of distributed generation over long periods of time. Combining these two modelling techniques allows taking advantage of each method's strength to obtain the necessary simulations over different timeframes. Their integration in a coordinated manner, through the development of a software platform, allows both visualizing the dynamic evolution of the system on a fine temporal scale, and planning for the lowest cost placement of renewable generators.

The ABM module presented in this paper is of most relevance to this thesis, as it describes how the physical configuration of the infrastructure with the description of the network assets and the way they are being used (through a description of their behaviours) come together. This model was developed so that it could support the examination of the system as a whole, where the impact of a change in one part of the system on other parts can be assessed thanks to the description of the individuals and their interactions with one another. The load at any point in the network can then be assessed thanks to the modelling at a fine level of detail, both geographically and temporally. This tool enabled scaling up of the studies of local electricity generation to the distribution level, providing greater insight into the impact of decentralised renewable energy generation on the grid. At the time this paper was written, there was no other ABM of distribution grid evolution including the physical infrastructure of the networks. Other existing ABM implementations used a "copper-plate" approach, where electricity could be added or removed at any location, that is there was no constraint from the network structure.

Because developing an ABM of the electricity distribution network is complex, especially as more entities need to be represented, the next step, as mentioned in the conclusion of this paper, was to look into modelling techniques for flexible and extensible ABMS using a modular or compositional approach.

To this end, MODAM was developed.

MODAM stands for MODular Agent-Based Model. It is a software framework that was built applying the concept of modularity to the implementation of ABMs for the electricity distribution grid, so that flexibility and extensibility of the models can be reached. This approach is novel to building agent-based models because the modular approach is not only for the software implementation, where a loosely coupled architecture is used with the separation of the model, the GUI, the database, the simulation engine, etc, but also for the model itself where the agents are decomposed into assets and behaviours. The asset and the required behaviours are dynamically brought together at runtime to form the customised ABM.

MODAM was first introduced in (Boulaire, Utting, & Drogemuller, 2013a) and was presented at the 2nd International Workshop on Software Engineering Challenges for the Smart Grid (SE4SG), at the IEEE International Conference on Software Engineering (ICSE), San Fransisco. A journal paper was then written that describes how MODAM is built (Boulaire, Utting, & Drogemuller, 2015b), especially detailing the dynamic agent composition approach taken to support the development of flexible and extensible large-scale agent-based models. It has been published in the journal of Complex Adaptive System Modelling.

Chapter 4 describes the dynamic agent composition method which is at the core of the MODAM implementation. The modular approach to building our ABMS application was chosen in order to allow flexibility and extensibility, inspired from component-based software engineering practices. The motivation for taking such an approach is described in this chapter and can be summarised as:

- Answering a need for flexible options handling, in terms of data sources, data models and simulation parameter inputs. This need arose from the client's requirements, which were that the application was to be tailored to their network for which data was available;

- Answering a need for extensibility of the model as new technologies would need to be assessed, and new ways of using them with additional behaviour options;
- Answering a need for both flexibility and extensibility to be able to answer a vast range of questions, and try different scenarios of possible futures, where alternative behaviour could be tried as well as different mixes of asset types.

By using the dynamic agent composition described in this chapter, it is possible to extend an ABM of a networked structure with ease, as well as having it flexible so that many scenarios can be created using large corporate databases. The agents, whose information relating to their asset properties or behavioural characteristics can be stored in independent components and populated with data, can be brought together at runtime into a coherent agent-based model. This linking of the components can be done manually; however, this would require a modeller to write code, which is limiting in terms of the type of users who can build models with MODAM. These would need to be able to code in Java, which is not desirable as the software is to be used on an everyday basis by planners within power utilities. To answer this requirement of code-free setup of simulations, this process was automated.

Chapter 5 describes this automation process. Three aspects support this process: the technology used to implement the software system, the way large-scale ABMs are specified using command-line scripts or GUIs, and the automation of the composition that brings the different elements together. These are described in this chapter, where the Module Manager that weaves the agents together at runtime is formally specified using Z. While implementing such a mechanism brings challenges especially for large-scale ABMs that represent a networked structure, it offers many benefits such as rapid model building and set up of a vast range of scenarios, in addition to facilitating reproducibility of simulations.

However, as the model grows and the number of possible actions and interactions increases because of additional agents, the speed of the simulation runs might start to slow down. Also, as the agents' decision rules become complex, especially when learning algorithms are implemented, the speed is expected to be further reduced. In order to support faster simulation runs, a parallel scheduler was implemented in a way such that it leverages the structure of the ABM.

This parallel scheduler is presented in **Chapter 6**. It is a fine-grained shared-memory parallel implementation that takes advantage of the directed graph structure of the network. The agents are grouped and executed on a multi-threaded machine, taking advantage of the structure which is an aid to the parallelisation. This chapter was written as a conference paper (Boulaire, Utting, & Drogemuller, 2013b) that was presented during the 1st Workshop on Parallel and Distributed Agent-Based Simulations as part of the Euro-Par 2013 conference, and was published in volume 8374 of the Lecture Notes in Computer Science series.

Having described the infrastructure supporting building the agent-based models, **chapter 7** presents two types of agents that have been implemented in MODAM: the electricity consumption of individual consumers using historical load data for the different customer types (residential, commercial and industrial) and the electric output of a rooftop photovoltaic system subject to weather variability with the inclusion of cloud data from the Bureau of Meteorology (BOM). Details of the implementation of the agents are given. Simulations are then run in the view to answer the question: "*What is the capacity of solar generation that would reduce peak demand at particular points in the network?*". The results are given at the transformer level whose load is the aggregation of all the connected consumers. These loads are compared to an equivalent 1kW PV system at that same location so that sizing of the PV systems to be installed can be performed. This chapter illustrates the use of MODAM for a well-bounded problem, where the space under study is limited, and the question is well-defined. It also shows that the agent-based M&S application can be extended with the addition of new agents or localised adaptation of existing ones. This paper was presented at the 2012 Solar Conference in Melbourne (Boulaire, Utting, Drogemuller, Abeygunawardana, et al., 2012), and was awarded best post-graduate paper at the conference.

Chapter 8, written as a journal article (Boulaire, Utting, & Drogemuller, 2014), is another illustration of the use of MODAM but within a broader context, where a much more complex input scenario was used to assess the impact of technology uptake. In this chapter, simulations of possible futures are created by mix-and-match of agents, and the simulations outputs analysed to gain an insight into how PV and EV might impact the grid in the future. One scenario was investigated from which four simulations were derived, to represent the possible trajectories of consumption. These simulations were run over two areas in Townsville, Australia, and highlighted the geographical implications of a variation in impact of similar behaviour. This chapter details the development of the input scenario and the implementation of the electric vehicle agents for which two charging methods were implemented. This highlighted how the way a technology is used might impact the state variable (e.g. the peak load) at any location on the grid. From the analysis of the simulations outputs presented in this chapter, planners are able to identify quickly, thanks to the visualisation capabilities of MODAM, the assets that are at risk of overloading, under certain conditions of use. These simulations are of significant benefit to planners to inform their decisions when planning the grid, especially as assets at risk of overload, which might need upgrading or replacing, can be identified individually. The results presented in this paper further emphasize the importance of modelling electricity flows at a fine level of detail geographically and temporally, to capture the spatial and temporal variability of load consumption identified in chapter 3. However, it contrasts with the results presented in chapter 3, where understanding the trajectories of load requirements over zones is not only important, but linking these loads to the assets on the grid is of additional importance, if these are to substantially inform decision-makers when planning upgrade and replacement of assets on the grid.

Overall, this PhD research program proposes a novel approach to model large and complex physical systems using agent-based modelling. It consists of using a compositional method to building agent-based models so that extensibility and flexibility can be achieved. It is especially developed for, and applied to, electricity distribution networks and used to investigate the impact of the uptake of new technologies on the grid.

Chapter 2: Literature Review

Some elements of this literature review are cited in the papers that form the chapters in this thesis. This format was chosen so that the papers remained as self-contained contributions to this thesis, and this chapter could still be read independently. Additional information is provided in this chapter (1) to set the case studies presented in this thesis in a broader context by introducing the electricity sector in Australia, (2) to introduce in more detail complex systems and the particular case of agent-based modelling as a technique for planning the distribution grid, (3) to present the different concepts of a compositional approach to modelling and simulation used in the software engineering domain.

Modelling and simulation are two terms often associated with one another in relation to describing a real world system and observing its behaviour on a computer. While there are many definitions for modelling and simulation (Ören, 2011), a simple one of both these terms can be that modelling is a way to capture the information of a system, while simulation consists in running the implementation of the model and see what happens (Maria, 1997). Bernard Zeigler, in (Zeigler, 1976), defines three entities that are concerned when it comes to modelling and simulation: the real system, a model, and a computer, where *modelling* represents the relationship between the real system and the model, and *simulation* the relationship between the computer and the model. These three entities therefore need to be considered simultaneously when building a modelling and simulation (M&S) application as they will influence each other in turn. This definition is the one followed throughout this thesis. The literature review presented in this section will describe in turn these three entities in the context of this research – however, as they are intrinsically related, references to one another throughout this section can be identified.

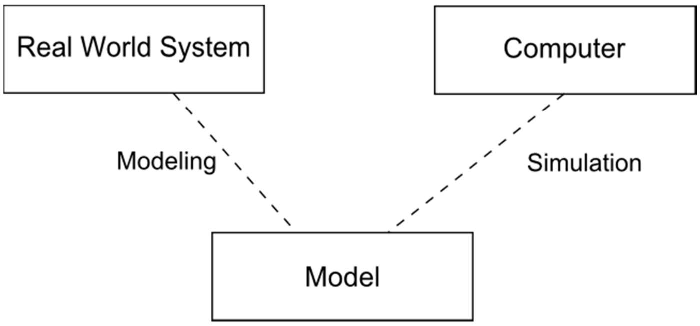


Figure 2-1 - Modelling and Simulation – Interaction of Real World System, Model and Computer. Source: (Zeigler, 1976)

In the context of this research, the domain of application is the electricity distribution grid and the impact of small-scale renewables and other technologies on its planning and operation. The following section therefore describes the electricity sector in Australia as well as the different energy sources and technologies relating to renewables. Section 2.2 will then describe the different modelling and simulation approaches that have been used in the electricity domain and why agent-based modelling is suitable for modelling the distribution grid. Section 2.3 will follow with the methods and tools specific to building agent-based M&S applications with an emphasis on what happens as the models become large. Section 2.4 will finally cover the different methods that are known in software engineering that can be used in the context of building large-scale ABMs to support the implementation of the model specific to our research domain.

2.1 ELECTRICITY IN AUSTRALIA AND MODELS AND TOOLS USED IN THE SECTOR

2.1.1 Electricity sector in Australia

The electricity sector is composed of many entities of different natures which are inter-connected. Four main categories of entities can be distinguished that play a specific role at the different levels of the electricity system:

- The entities dealing with the **physical components** of the sector, which include the generating companies, the transmission network service providers and the distribution network service providers;

- The **energy market**, dealing with the dispatch of energy to answer the energy demand and the fluctuation of energy prices;
- The **regulatory bodies** which enforce the rules established by the different governmental commissions;
- The **'prosumers'**, who consume electricity and may produce electricity if they play a role as distributed generators.

While these entities have distinct roles, they are interrelated at different stages of the process. A brief description of these actors is given below.

The Generation and Delivery of Electricity

The generation and delivery of the electricity under the dominant central generation model is often ensured by different bodies and follows many steps that can be summed up as follows (Lilley, Szatow, & Jones, 2009; Oncor, 2012; Wikipedia, 2012b):

- The electricity is generated at a generating station or power plant, which can be powered by fossil-fuel or renewable energy;
- This generated electricity is then converted from low to high voltage (132kV-500kV) so that it can be transported more efficiently over long-distances through the transmission network. Some customers, called transmission customers, can access electricity at this stage, at 132kV or 235kV;
- Once sufficiently close to the consumer, the electricity goes through a step-down converter to a lower voltage for supply to the distribution network;
- The electricity finally reaches the different consumers after having its voltage reduced further through a substation. Depending on the nature of the consumer, the voltage is reduced to 33kV-66kV for subtransmission customers, 11kV for primary customers and 240V for secondary customers.

Generating companies, transmission network service providers and distribution network service providers collaborate at the interface of their systems, ensuring the safe and reliable delivery of electricity.

The Electricity Market

Following the deregulation of the electricity market, the National Electricity Market (NEM) started operating in December 1998 as a wholesale market to supply electricity to retailers and end-users in Queensland, New South Wales, the Australian Capital Territory, Victoria and South Australia. The NEM operates over a distance of around 5000km, the world's longest interconnected power system (Australian Energy Market Operator (AEMO), 2010).

In order to manage the NEM and the gas market, the Australian Energy Market Operator (AEMO), previously NEMCO, was established in July 2009. One of its key aims is to provide an effective infrastructure for the efficient operation of the wholesale electricity market, to develop the market and improve its efficiency and to coordinate planning of the interconnected power system (Australian Energy Market Operator (AEMO), 2010). As such the electricity market is fully interconnected to the generation and supply of electricity, see Figure 2-2.

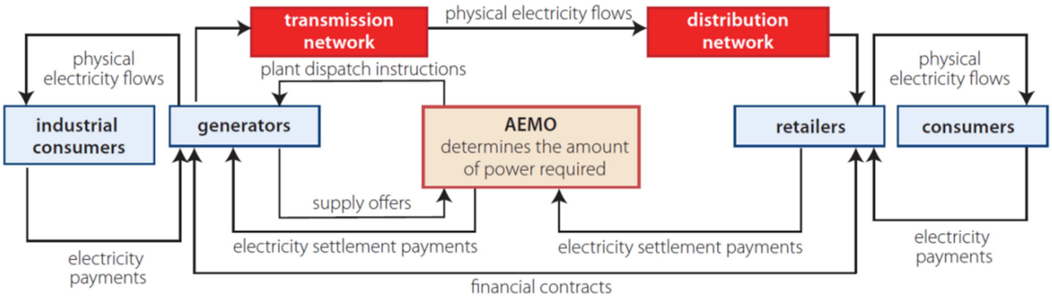


Figure 2-2 - Electricity market. Source: (Cuevas-Cubria et al., 2010)

The Australian Energy Regulator

The Australian Energy Regulator (AER) was established in July 2005. The AER regulates the wholesale electricity market and is responsible for the economic regulation of the electricity transmission and distribution networks in the national electricity market (NEM) (Australian Energy Regulator, 2010). The AER's regulatory functions and powers are conferred upon it by the national electricity law and the national electricity rules.

While the price of energy is dependent on the energy market, the revenues for the distribution and transmission companies are regulated by the AER. These revenues are calculated following a set of rules which are dependent on the network and the levels of service (Ergon Energy, 2011).

The 'prosumers'

The term 'prosumer', coined in 1970 by futurologist Alvin Toffler, has been used (Sarvapali, Perukrishnen, Alex, & Nicholas, 2012) to describe actors in the market place that not only consume but also participate in the production of goods. This type of actor can now be found in the electricity sector. While in the past the electricity produced by large generators was distributed down through the network and used by the consumer, now, with the introduction of decentralised generators (DGs), such as rooftop solar panels, points of consumption are also becoming points of production.

Three main types of 'prosumers' can be distinguished: industrial, commercial and residential. The distinction between these classes lies in their sector of activity, but also reflects the amount as well as the patterns of the loads consumed. With the large scale introduction of DGs, changes within these classes are expected. Indeed, not only can the consumption be reduced at times when the generators are producing but in some cases their production of electricity can exceed their consumption, leading to electricity flowing back from the consumer to the grid. Also, it is expected that as the energy market opens up to small producers, these 'prosumers' might begin to have an impact on the electricity market. The level of such an impact is yet to be understood.

As can be seen, the different actors in the electricity sector have distinct functions and responsibilities. However, while acting independently from one another, they are highly interconnected. It is therefore expected that changes in the functioning in one part of the system will impact on other parts. Such interconnection is at the heart of the complexity problem of this research.

2.1.2 Energy sources and technologies

The majority of electricity currently produced in Australia is generated using coal (69% in 2011-2012), and natural gas (19.26% in 2011-2012), with the remaining 9.46% produced by renewables, as shown on Figure 2-3. While still a minority, the share of electricity generation by renewables has been increasing slowly over the last 15 years (Figure 2-4) with the introduction of biomass, biogas, solar and wind power generation, while hydro had been used over a few decades.

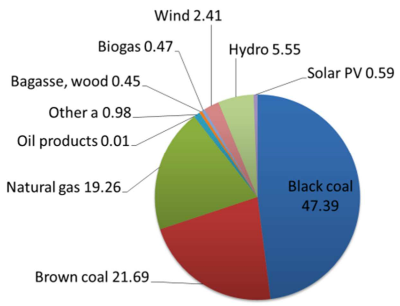


Figure 2-3 - Australian electricity generation by fuel (in %), 2011-12; includes multi-fuel fired power plants. Source: (BREE 2013, 2013).

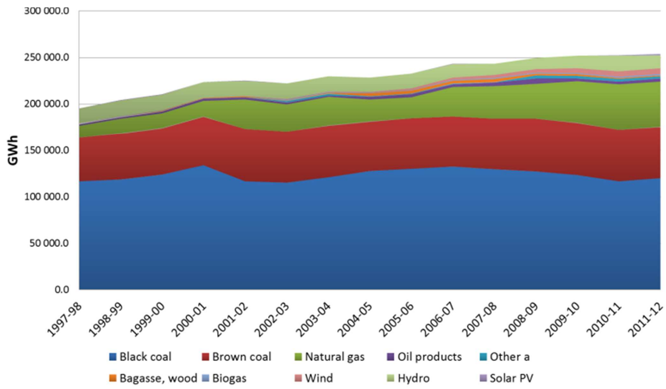


Figure 2-4 - Australian electricity generation by fuel, 1997-2012. Source: (BREE 2013, 2013).

Following the rising demand in electricity, generation capacity has been increasing in Australia over the years with the opening of new power plants. In 2007-2008, Australia's electricity generation capacity was about 49 GW (Cuevas-Cubria et al., 2010), and reached around 56 GW a year later, 2008-09 (World Nuclear Association, 2010). The capacity is expected to continue increasing with the creation of new projects.

While generators using non-renewable energy sources are still being planned, more projects are focussing on the integration of renewable energy generation. Different types of technologies are available which can be integrated at various levels in the electricity delivery hierarchy (high-voltage centralised generation to low-voltage distributed generation). Australia already has a few implemented renewable energy projects, many of which are using solar energy generation. With its high solar radiation average, Australia and in particular Queensland is a very good candidate for such technology.

As an illustration, the largest solar power station, located at Singleton, New South Wales is a 400 kWp (kilowatts, peak) photovoltaic array. Smaller arrays have also been used for some time, mainly in the remote areas of Australia where solar power is abundant and cost-competitive with diesel power. At a lower level in terms of capacity generation, solar photovoltaic (PV) panels have increasingly been installed on individuals' rooftops, bringing Australia's solar PV capacity to around 3,017MW at the end of October 2013, compared to 1,450 MW at the end of February 2012 and to an initial capacity of 23 MW in 2008 (AEMO, 2012; Australian PV Institute & Australian Renewable Energy Agency, 2013).

Advances in the efficiency of the technology as well as the effect of the different targets and incentives set by the government have allowed generation of electricity using solar power to become more widespread in Australia. Such progress is very encouraging, meaning that the target of 20% of renewable penetration by 2020 is attainable. However, research has shown that with penetration rates of variable generations increasing over levels of 15% to 20%, it can become increasingly difficult to ensure the reliable and stable management of electricity systems, depending on the electricity system in question (International Energy Agency, 2011). Some countries in Europe have managed to reach quite high percentages of variable renewable generation (over 10% for wind for example), which was achieved thanks to cross-border transmission links (Kirby & Milligan, 2008). Australia, being isolated from other countries, needs to deal with the technical challenges in a different manner.

In conclusion, when modelling the integration of renewable systems in the current grid, it is crucial to consider their specific technical constraints but also the interconnection between the different components of the sector as it is expected that one change in one part of the system will impact another part.

2.1.3 Models and tools applied to the electricity sector

With such a vast sector, it is expected that a large number of analysis types are required to answer the different needs relating to the electricity distribution. Consequently, a large number of tools can be found that vary in their purpose, their nature and their use. This section introduces some of these tools and models, targeting more specifically those for the analysis of systems dealing with the introduction of renewables and with network constraints.

Tools – domains of application, technical and geographical characteristics

A large number of tools, developed for various analyses in the electricity sector are available. For example, in their review (Foley, Ó Gallachóir, Hur, Baldick, & McKeogh, 2010), Foley et al. have described 7 models applied to electricity systems; Connolly et al. have also identified 68 tools (Connolly, Lund, Mathiesen, & Leahy, 2010) for analysing the integration of renewable into various energy systems, 37 of which are described in more detail. Many more tools are available that can be found on the internet, for example (Powertech Labs Inc, 2012; ROAM Consulting, 2012), without counting the ones developed internally by the different electricity authorities.

In (Connolly et al., 2010) the authors compared 37 tools according to different criteria, such as the type of application and the sectors they were developed for, the number of users they have, the type of models used within the tools, as well as the level of renewable generation penetration that these tools allow analyses for. The tools presented in their paper are very diverse as indicated by the many values the different criteria take. Indeed, the tools application domains range from performing analysis of systems at the building level (BCHP Screening Tool), to the district level (BALMOREL (Ravn, 2012)) and to assessments at the national level (INFORSE). Also, the tools can be used for the electricity sector only, or for a combination of the electricity sector with the heating and the transport sectors. Finally, tools such as Homer (HOMER Energy, 2012) allow simulating 100% electricity using renewable energy, and others such as EnergyPLAN (EnergyPLAN, 2012) or H_2RES (H2RES MODEL, 2009) can also simulate 100% renewable but for energy. Being able to classify these tools according to such criteria is useful when looking at answering a particular question to a problem. Within the context of this research, the inclusion of the technical constraints of the network in the analysis is a criteria of interest when assessing modelling tools.

In (Connolly et al., 2010), the authors only identified 4 tools out of the 37 detailed in their review, as techno-economic (COMPOSE (Connolly, 2009), EMCAS (Argonne), HOMER (HOMER Energy, 2012) and energyPRO (EMD International

A/S, 2012)); however none of these tools consider the distribution grid in their analysis. By techno-economic, the authors mainly meant that the technical characteristics (such as inverter capacity or PV size) of the renewable systems were used to inform the analysis. Only EMCAS, identified in their review, considers the technical constraints of the network; however, the constraints are at the transmission level and not for the distribution network.

Other tools apart from EMCAS, were further identified in the literature, which use transmission network technical constraints. These tools are generally in the domain of electricity systems modelling and are used to manage electricity demand and generation, the systems and the trading of electricity. This type of analysis is generally done at the national level as the generation and dispatch is traditionally done from centralised generators. Such tools include PowerACE (Genoese, Sensfuß, Weidlich, Möst, & Rentz, 2005), Prophet (Intelligent Energy Systems, 2011), EMCAS, Genersys (Batten & Grozev, 2006) and AURORAxmp, PLEXOS, GTMax, UPLAN, WASP, WILMAR as described in (Foley et al., 2010). Most of these tools can integrate the analysis of renewable generation in their analysis; however, the renewable systems considered are large-scale renewables such as large wind farms or large-scale solar systems. Different analyses can be performed using these tools that take into account the carbon emission as well as the climate dependency of demand and supply, providing companies with a way of making decisions when investing in new generation and transmission, as is the case for Genersys for example. While some characteristics of renewable generation are taken into account in those systems, such as intermittent generation over the days and the years, the geographical location of the systems is not considered. These analysis systems treat the physical network as a “copper-plate” as if electricity generation happened in a similar manner at any given location over the network. Consequently, the technical constraints included in these analyses are those occurring at the high-voltage level over very large regions (state-wide for example with interconnectors’ constraints only) but do not take into account the very specific characteristics of the areas of production and the local constraints on the network.

Another aspect of interest regarding the tools is their ability, or inability, to extrapolate information used for analyses at a fine level of detail to higher levels. Such characteristics would allow for example understanding the impact that large introduction of small-scale renewables has on the system as a whole. Indeed, while site-specific analyses are available when performing feasibility study for the installation of renewables, these analyses are not translated to higher dimensions in the analysis spectrum. As an illustration, ROAM's Consulting (ROAM Consulting, 2011) uses historical data from weather stations located over Australia in their modelling of wind and solar generation. This type of modelling is location-specific and therefore appropriate when looking at potential locations for investment options of large-scale renewables. The site variability in terms of its potential for generation due to weather and topological characteristics can be critical especially for investors who want to optimise their return as well as minimise their risk on investment. Such detailed analysis is therefore appropriate and available. However, when including this information into their whole of NEM analysis, it seems that this information is then aggregated at the state level without looking at the technical constraints over the network. There is a gap between site specific and nation-wide planning of renewable generators. In a similar manner, installations of renewables at the low-voltage level of the network do not include analysis of the network.

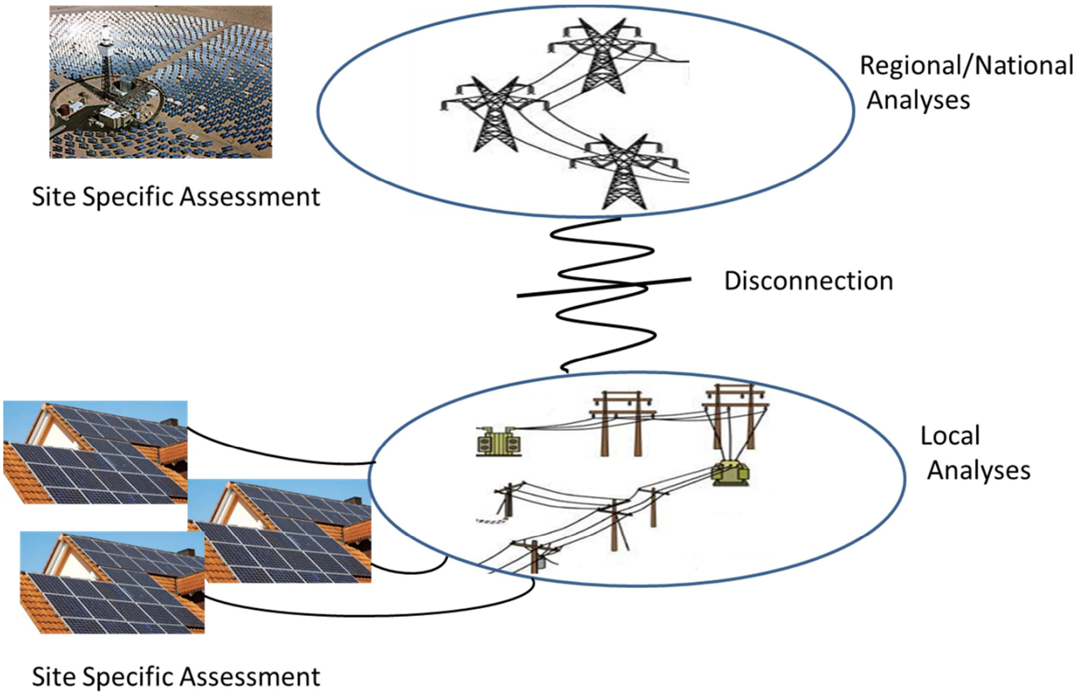


Figure 2-5 - Levels of electricity systems modelling. Site specific analyses are done both for large-scale and small-scale renewable systems but there is a disconnection between house level, distribution network up to transmission network.

As can be seen here, a wide variety of tools are available to perform analyses for the introduction of renewables at different levels of details for different applications within the electricity sector. Despite this wide range of tools and models, no tool has been identified in the literature that supports analyses for the large-scale introduction of renewables taking into account the technical constraints at the fine level of detail (e.g. geographical and distribution network constraints) so that the impact can be assessed at the system level. Also, while site-specific analyses assessing the benefit of installing a renewable generator are available, this information is not transferred into other analyses that are assessing the integration of renewables at higher levels in the system, such as the state level for example. Such disconnection in the analyses available is one of the reasons that prevented seeing the effect that many DG installed in some areas would have on the network (Hall, 2011). Figure 2-5 illustrates such disconnection in terms of the levels at which analyses are available within the electricity system, when considering the installation of renewables.

Methods and models used in these tools

In order to perform the analyses in the tools introduced in the section above, numerous modelling techniques are used. These can be classified for example as: simulation, scenario, equilibrium, top-down, bottom-up, operation optimisation or investment optimisation models (Connolly et al., 2010). These types of categories are broad, meaning that within each of them there exist more sub-categories. Also, modelling techniques can be combined in one tool in order to answer the level of complexity required. As an example, BALMOREL was classified in (Connolly et al., 2010), as using all the modelling types mentioned above, except for a top-down approach.

In the context of this study, two aspects need to be considered when looking at modelling the integration of renewables on the grid. The first aspect concerns the modelling of the electricity consumption and the second one is the electricity production from the decentralised generators; these two aspects are however interrelated as the place of consumption and production may be the same. The following paragraphs review modelling techniques that are used to model electricity consumption and generation from renewable generators.

Electricity consumption modelling

In their paper (Swan & Ugursal, 2009), Swan et al. review many modelling techniques that have been used when calculating end-use energy consumption in the residential sector. The techniques presented are separated into two main categories: top-down and bottom-up approaches. Top-down approaches are usually based on historical aggregated data, losing explicit representation of end-uses, while bottom-up approaches consist of aggregating individual consumption up to different levels, ranging from individual houses to regions for example

Within the context of this study, bottom-up approaches seem the most appropriate methods when modelling energy demand, as the framework aims at knowing the demand at different levels of granularity within the system. Bottom-up approaches include statistical models with regression, conditional demand analysis and neural network, or engineering models with population distribution, archetype and sample models (Swan & Ugursal, 2009). As an example, (Mihalakakou, Santamouris, & Tsangrassoulis, 2002) used neural networks to model the energy on an hourly basis, and have obtained very good results. However, this model is specific to a given case study as it is dependent of the input data from a single house, using two parameters explaining energy consumption (air temperature and solar radiation); such a model would need to be recalibrated for every type of house. Other methods such as conditional analysis and regression analysis have been applied to bigger datasets, with different types of houses, and have not only allowed identification of the coefficients of interest that explain the relationship between energy consumption and different parameters but also qualified this relationship (Douthitt, 1989; Ndiaye & Gabriel, 2011).

Often, the data used for electricity consumption analyses is quarterly billing data or yearly totals data. However, as in the case of the neural network example mentioned above, time-series data can also be available. In this case, other types of analysis can be used such as time-series analysis studying lagged-correlation or regressions using ARIMA methods, for example. While it is important to quantify the effect of different factors on the electricity consumption, it can also be important to see the patterns of consumption within the different times-series in order to group premises that have similar patterns. Such an approach can be especially interesting when looking at modelling electricity consumption over different levels of aggregation within the network and when limited information is known about the type of premises these records are coming from (e.g. the characteristics of the household in terms of inhabitants and appliance usage). This type of analysis can become important especially when predicting consumption in a distant future when trends in households or appliances are estimated without having actual detail on the individual characteristics. Many clustering methods on time-series can be used (Riedy & Partridge, 2006) depending on the nature of the data as well as the purpose of the study.

Combinations of the different methods introduced here can also be used so that the different needs of the analyses can be met. Time-series analyses, clustering and regression analyses have been used for the analysis of the data provided by Ergon. Such analyses allowed the available data to be understood and support its reuse within the different parts of the framework.

Electricity generation and dispatch modelling

When considering the electricity generation from renewable systems, the modelling methods that are needed have to consider the variability in the generation of electricity. Consequently, a stochastic approach is often used that can predict the level of output from intermittent generation units, using weather conditions at site-locations (ROAM Consulting, 2011). A distinction, however, has to be made between large-scale and small-scale generation, as the techniques employed may differ.

For analyses of large-scale generators, the outputs of renewables are used in a similar way as for traditional generators when dispatching electricity. Indeed, such approximation can be applied because the output of the generators can be rather well predicted over medium time periods thanks to more and more accurate weather predictions at low levels of detail (when dispatching in the market, the bidding happens 24 hours in advance which can be quite accurately predicted). Also, even in case of a sudden unexpected weather event over a short period of time, the variation in the generator output is not as dramatic as for small-scale generators because the catchment areas are rather large and a smoothing phenomenon over the area of generation can be observed. In that case, electricity system modelling “is generally carried out using some form of stochastic (random) programming, which involves minimising an objective function subject to the expected cost of electricity dispatch subject to a number of constraints including availability and operational characteristics of generating plants, licensing environmental limits, and fuel costs, contractual obligations, operator and transmission constraints." (Foley et al., 2010). Further to this, there are two main approaches to modelling electricity systems which are *stochastic optimisation* and *dynamic programming* methods. This type of modelling was applied in tools such as AURORAxmp or GTMax. GTMax applies "a simultaneous optimisation of chronological hourly loads for one week to optimise the scheduling of hydro/thermal power plants taking into consideration transmission constraints, ramp rates, system hydrology, scheduling of maintenance, bilateral contracts, and opportunities for trading on the spot market" (Foley et al., 2010). As can be seen here, while a stochastic approach to the expected output of large-scale generators needs to be used, the same principles when dispatching electricity onto the network apply as in the case of traditional generators. Also, once the produced electricity reaches the transmission network, no more technical constraints specific to intermittent generation are observed; only the level of production that cannot be controlled as in the case of a thermal plant, for example, varies.

When considering electricity generation from small-scale generation, these types of criteria do not apply and therefore modelling techniques specific to small-scale generation need to be applied. In that case, analyses specific to the DG and the location at which it is going to be installed are used. For this, various tools such as PVWatts (National Renewable Energy Laboratory, 2012) can be used. These take into account the specificity of the solar panel and apply the weather conditions for a typical year. Half-hourly estimation of PV output are given, which are averages for a given weather condition; no randomisation of the weather event is used in the software, but this can off course be added later by hand, if required. When scaling up to predictions at regional level, the same types of output are used but scaled up. In (AEMO, 2012), AEMO describe the modelling method used to calculate the output from solar PV over a region. The method employed uses the daily energy generation from a 1kW system in a given capital city, combined with the monthly averages for each capital city, calculated using PVWatts (National Renewable Energy Laboratory, 2012). This results in an adjusted daily electricity generation for each month for a

1kW system that is then multiplied by the PV capacity of the area of interest in order to obtain the overall regional generation. At times of maximum demand, a different analysis can be performed which uses ½ hourly data recorded for a sample of solar panels, and multiplying it by the region's forecast installed capacity.

While this type of analysis is a good approximation of the electricity generation from solar panels over a region, it only gives averages over typical days within a year. It is important to consider variation within days when managing the distribution network, because variations over a short period of time can lead to the failure in the safe delivery of electricity which is a constraint that networks need to fulfil, regardless of the type of generators.

Planning for the grid

When planning for the future electricity distribution grid, often, conventional optimisation techniques are used such as branch and bound techniques, or heuristic methods (e.g. particle swarm optimisation). These techniques look at minimising the cost of infrastructure that can include the installation, maintenance and reliability costs, given different constraints. These include constraints on voltage and current at each network bus, the size of the different assets to be installed (Arefi & Ledwich; Gomez et al.; Marshall, Boffey, Green, & Hague, 1991; Ziari, Ledwich, Ghosh, & Platt, 2012). These techniques are mainly used from the point of view of a single decision maker which in this case is the grid operator and that assumes that an equilibrium point will be reached when all the participants reach a common ground. While this approach is appropriate when the energy system is mainly managed by one representative body, it becomes much more complex when participants can influence the system with installing decentralised generators and managing them according to their own objectives.

2.2 MODELLING AND SIMULATION FOR PLANNING THE DISTRIBUTION GRID

The electricity sector has been transitioning from a centralised to a decentralised system over the last ten years in Australia, with a greater number of installations of small-scale renewables. This trend is expected to continue as other new technologies such as small-scale storage systems (batteries) and electric vehicles are coming on-line and are expected to see their numbers increase. As these technologies bring changes in terms of how, where and when they consume and produce, their impact might be exacerbated as their owners seek to maximise their interest. Further, in some cases, unexpected outcome at the system level might be observed resulting from the different actors' conflicting interests. Then, a rethink of many aspects of the system will be needed, which will not only involve the physical integration of the new system units over the network, but also how they will be integrated into the system and interacted with, which calls for a new way of studying the grid (Elliott, 2010). Complexity science has the potential to provide insight and analytical power to plan and manage the electricity grid: "*The energy system involves many multi-level multi-goal interacting systems, with many possible actual outcomes. This is what complex systems science aims to help decision-makers deal with.*" (Elliott, 2010).

This section gives an overview of complex systems and complex networks in the context of the electricity grid, along with the different modelling techniques that have been used for it. Agent-based modelling, which is a method used to describe complex systems is then introduced along with how it has been used in the electricity sector.

2.2.1 Complex systems

A complex system is defined by (Simon, 1962) as "*one made up of a large number of parts that interact in a nonsimple way.... In such systems... given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole*" (p. 468). Not only these systems are characterised by the large number of components, they are also characterised by the diversity of their components, their relationships and their interactions. Further, such system can have a multiplicity of possible outcomes due to its capacity to choose, to explore and to adapt (Nicolis & Rouvas-Nicolis, 2007). Finally, complex systems exhibit properties of emergence which notion may be subdivided as "weak" or "strong" emergence as defined by Bedau in (Bedau, 1997). Strong emergence is defined as nominal emergence which involves downward causation, where the resulting phenomena cannot be reduced to the system's constituent parts. In contrast, weak emergence results from the system's global behaviour that can be derived from the interactions at the micro-level, which however cannot be explained simply because of the complicated relationships of these micro-level interactions. In summary, complex systems are made of elements whose characteristics are represented at the individual level, along with the connections between them, and through building the system from the lower (micro) level to a higher (macro) level, a phenomenon of emergence can be observed.

Complex adaptive systems, which are a special case of complex system, are distinguished by their ability to adapt, "*to self-organize and dynamically reorganize their components in ways better suited to survive and excel in their environments*" (Macal & North, 2006) at the individual or collective level resulting from experience. They have a range of scale mechanisms described in (Macal & North, 2006) such as "*allowing to form groups (aggregation), invalidate simple extrapolation (Nonlinearity), allow the transfer and transformation of resources and information (Flows), allows agents to behave differently from one another and often leads to the system property of robustness (Diversity), allows agents to be named and recognized (Tagging), allows agents to reason about their worlds (Internal models) and allows components and whole systems to be composed of many levels of simpler components (Building blocks)*".

The distinction between complex systems and complex adaptive systems lies in the fact that in the latter, the systems have the capacity to respond to changes in their environment, learn from them and adapt. The term complex system encompasses complex adaptive system as well as many other theoretical frameworks which are used to investigate questions about evolving and adaptable systems. Complex systems have a vast number of domains of applications such as ant colonies, human economies and social structures, climate, nervous systems, cells and living things, including human beings, as well as modern energy or telecommunication infrastructures (Wikipedia, 2012a); they are also studied by many areas of natural science, mathematics, and social science. Fields that specialize in the interdisciplinary study of complex systems include systems theory, complexity theory, systems ecology, and cybernetics.

The electricity distribution grid: a complex system

Electric power systems can be defined as complex systems (Rinaldi, Peerenboom, & Kelly, 2001). The electricity distribution grid, which is a component of this infrastructure, can also be defined as a complex system. Indeed, the electricity grid is made of a very large number of elements that are interconnected, and whose interactions can result in very different outcomes for the grid depending on how they are behaving. These elements are not only the physical assets that are making up the grid, but also the people interacting with it. These interactions can vary depending on the environment conditions at any given time, whether it is on a minute-by-minute basis because of their fluctuating patterns in consumption, or over longer periods of time where the introduction of new technologies or policies can influence the grid configuration and in turn the load flowing through the network.

As an illustration, small-scale generators, such as rooftop PV, are being installed in an ad-hoc manner over the network according to the household owners' decision. While the installation of these generators can be prevented by the distribution companies in case of potential problems on the network (Hall, 2011), their installations cannot be forced upon the customers even if this could help solve some network issues. As an example, these could help at locations where consumption patterns show peak load times coinciding with PV generation times or where batteries' usage could shave the peak. In addition, the small-scale generators output their electricity surplus directly into the distribution network, which is not at a central point but scattered over the network. This implies that while there might be some smoothing of electricity generation over the installed base of these units (through the passage of clouds) the distribution grid has to control the sudden changes in generation at each point as they are directly connected to the grid. The grid therefore has to adapt to these different changes over time. Regardless of what is happening on the grid, it needs to adapt so that the supply matches exactly the demand, under the constraints of each element of the system. Finally, the output into the grid cannot be easily predicted as only the surplus of electricity generation gets fed into the grid, meaning that the output not only depends on the production from the natural element conversion, but it is also highly dependent on the behaviour of the inhabitants which attitude towards consumption might change depending on different factors, such as the benefit of feed-in tariffs. Finally, with the promise of more and more affordable batteries, these could become major actors on the variation of the network flows depending on the control algorithms chosen.

From these examples, we can see that the many and different types of actors that are composing the electricity grid have interdependent relationships which resulting interactions might lead to an emergent behaviour at the system level. The electricity grid can consequently be defined as a complex system. The description of the connections and interactions of the different elements can further be formalised using mathematical graphs, or complex networks, which is one of the fields of complex system science.

Complex networks

The topology of the interactions of many real networks have been studied for many systems such as biological (Camacho, Guimerà, & Nunes Amaral, 2002; Solé & Montoya, 2001), social (Watts & Strogatz, 1998) and communication (Vázquez, Pastor-Satorras, & Vespignani, 2002) systems. The power grid which is composed of many non-identical components that are connected to one another is also a complex network (Albert, Albert, & Nakarado, 2004). It can be mathematically described as a graph $G = \{V, E\}$, where the nodes V represent the different components (e.g. transformers, lines, switches) and the connections between them are the edges E. In addition, it is a spatial network, as the nodes have a precise location in a 3D Euclidean space.

Different metrics are commonly used to describe a network (Boccaletti, Latora, Moreno, Chavez, & Hwang, 2006). These include

1. node degree, degree distribution and correlations,
2. shortest path lengths, diameter and betweeness,
3. clustering,

4. motifs,

5. community structures, and

6. graph spectra

Depending on the value of these metrics, different types of networks have been defined (e.g. fully connected networks, ring lattice networks, random networks, small world networks, scale-free networks) and are commonly found in many applications.

As described in (Albert et al., 2004) who studied the structure of the North American power grid in respect to its structural vulnerability, the power grid is a scale-free network, where the majority of nodes have low degrees but for which some have high-degree nodes (hubs), with the degree distribution following a power law, see Figure 2-6. A scale-free network has properties such that 1) they are more robust against failure, that is the network is likely to stay connected after the removal of a random chosen node, in comparison to a random network, 2) they are more vulnerable against non-random attacks, especially as high-degree nodes would be targeted, which would lead to the rapid disintegration of the network, 3) they have short average path lengths. These properties applied to the power grid mean that it is resilient to the random loss of nodes, but vulnerable to attacks targeting the high-degree hubs.

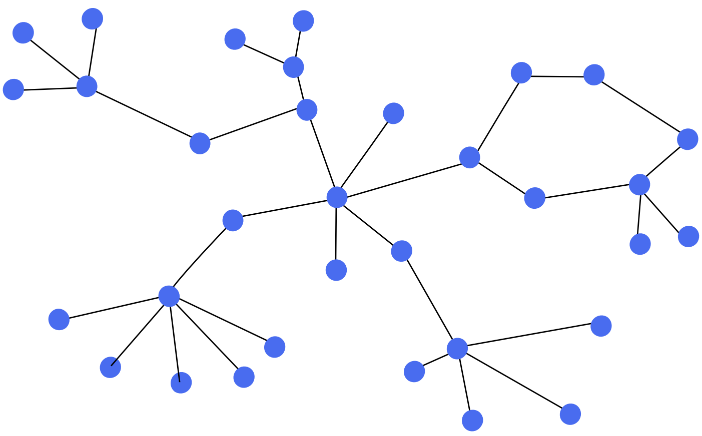


Figure 2-6 - Scale-free network, characterised by a majority of low-degree nodes and a few high-degree nodes.

As this research aims at modelling the underlying physical structure of the distribution grid in addition to the behaviours of the different actors impacting it, representing it as a complex network is useful. However, the method of choice for this study is not complex network, but elements from this field are going to be used, especially when improving the speed of the simulation which scheduler can be implemented taking advantage of the network structure.

2.2.2 Modelling techniques for complex systems

Different traditional approaches are commonly used for modelling complex systems. These include but are not limited to systems dynamics, discrete-event simulations, participatory simulation, optimisation, statistical modelling and risk analysis (North & Macal, 2007). These techniques have different properties that are most suited to some applications. For example, systems dynamics is used to model dynamic processes which change constantly over time; it is also useful for identifying important variables and causal linkages in a system. Optimisation is used when selecting a best solution from a set of available alternatives that can be expressed in mathematical expressions. Statistical models aim at explicitly quantifying a relationship between a dependent variable (output) and a set of independent variables (input) that will explain implicitly the system’s internal structure and causal processes. As can be seen in these examples, the nature and goal of these techniques is very different and could be used to represent some of the parts of the electricity system being modelled in this research.

While complexity science can be used to represent the electricity grid, looking at the system from a new perspective where a whole-of-system approach is taken, it does not aim to substitute any of the well-defined classical tools and methods in the fields. In addition to the methods mentioned above, other approaches dealing with power flow, grid unbalances issues, etc. can be used in combination to the selected method within the complex system representation chosen in this research.

Agent-based modelling, which is probably currently the most widely used technique amongst complex systems techniques, was selected for this research. It has properties best suited to the modelling of large systems, and can also benefit from the integration of the techniques described above to complement its modelling information (North & Macal, 2007). The following section introduces agent-based modelling and its suitability in describing the electricity sector.

2.2.3 Agent-based modelling

Agent-based modelling found its origin in the study of complex adaptive systems. This approach can be traced back to the 1960s as von Neuman introduced the theory of self-reproducing automata (von Neumann, 1966), which was further popularised by Conway with "The Game of Life"(Conway, 1970), and then Schelling's segregation model (Schelling, 1971). As computational power increased, cellular automata became more popular and evolved, and new paradigms emerged, one of which is agent-based modelling. Agent-based modelling was originally applied to biological systems (North & Macal, 2007), but now has a very large number of applications such as the study of social systems, financial markets, risk analysis of infrastructure networks and is applied over many domains such as the water (Linkola, Andrews, & Schuetze, 2013; Moglia, Perez, & Burn, 2010) or the electricity sector (Argonne; Batten & Grozev, 2006; Genoese et al., 2005; Zheng, Meinrenken, & Lackner, 2014), as well as fire applications (Shi, Ren, & Chen, 2009; Tang & Ren, 2008).

Agent-based models (ABM) can be defined as a class of computational models that describe autonomous and interacting agents, whose aim is to assess the system as a whole, which is greater than the simple sum of its constituent parts, (North & Macal, 2007; Wikipedia, 2011). These agents can be individuals or groups of entities such as an organisation. Through the description of the elements composing the system and the relationships between them at a fine-level of detail, ABMs allow capturing the dynamics of complex systems and complex adaptive systems. Agent-based modelling is becoming increasingly widespread; one of the reasons for its popularity being that we live in a complex world, which in some instances is becoming more complex through human intervention. Indeed, many of the systems we need to analyse are complex (e.g. biological systems) or becoming more complex (e.g. the electricity system), where many interdependencies need to be taken into account. Thanks to the increasing power of computers and the better access of organised data at fine levels of granularity, systems that we previously were not able to model adequately can now be studied through the development of large-scale micro-simulation models (Macal & North, 2006). In addition, the popularity of ABMs might be further explained by their ability (House-Peters & Chang, 2011):

- “to incorporate both spatially and temporally explicit data,

- to model bidirectional relations between individual human agents and the macrobehavior of the social or environmental system being modeled,
- to capture emerging patterns at higher scales of the system that result from interactions at lower levels, and
- to blend qualitative and quantitative approaches”

Characteristics of Agent-Based Models

Agent-based models, also known as individual-based models, are composed of individual agents. While there is no universal agreement on the definition of an agent, most modellers share the same view regarding the properties of an agent. (Macal & North, 2005) describes the properties and attributes of agents in a practical modelling purpose as having the following attributes:

- Agents are autonomous and self-directed,
- Agents are modular or self-contained, and
- Agents are social, interacting with other agents.

They can also have additional properties such that:

- They live in an environment,
- Have explicit goals that drive their behaviour,
- Have the ability to learn and adapt their behaviours based on their experience.

These agents can be specified at various scales, defining the granularity of the model. As they are autonomous, decision-making heuristics, learning rules or adaptive processes have to be determined in the model as well as an interaction topology. Such properties make using agent-based modelling for the electricity network interesting. Indeed the different components making up the network can be defined as agents, with the network topology captured through the model topology and the agents can behave independently as they would in the real world.

Other criteria used to define an agent-based model are their capacity for analysing self-organisation, adaptation, and networked causality. These three criteria find justification in the context of this research. Self-organisation – decentralised generators, such as solar panels, are being installed by household owners on a voluntary basis, leading to a self-organised system. While the distribution system provider can prevent some installations in the event of technical infeasibility, it has no control in general terms over the evolution of the solar panels installations on the network; it cannot impose the installation of units on some households. Adaptation – with the availability of new sources of electricity (rooftop solar PV, small wind-turbines) and small-scale storage (batteries), consumers are changing their behaviours to optimise the return on their investment within the policy environment; this in turn has an impact on the load patterns at different levels within the network. Following this, technical adjustments over the network are required over time as the distribution network provider needs to update its system in response to the variation in electricity flows, and this happens over time as the need rises. Networked causality – new sources of electricity generation within the network will lead to changes in electricity flow and patterns over the different parts of the network. As such, load variations at one point of the network will impact other parts of the network leading in some cases to over or under-voltage issues.

Additionally, Macal draws a list of criteria (Macal & North, 2005) as to when thinking in terms of agents can be beneficial. Some of the listed criteria that are of importance in this context are:

- "When it is important that agents have a spatial component to their behaviors and interactions
- When scaling-up to arbitrary levels is important in terms of the number of agents, agent interactions and agent states
- When the past is no predictor of the future because the processes of growth and change are dynamic"

The spatial representation of the agents is important as the level of output of the solar panels is highly dependent on their location in terms of solar exposure; their impact over the network is also important because of the relation of the solar panels to one another. In some cases, high penetration levels of rooftop solar photovoltaic create reverse power-flow and voltage-rise, when back-feed is greater than consumption on parts of the low-voltage distribution networks. While impact of increasing the number of decentralised generators (solar panels or batteries) on the system has started to be observed, it is not fully understood how it pans out when reaching larger penetration rates. Also, other technologies such as electric vehicles are expected to be impacting the grid; however, due to lack of experience, their impact is not yet understood.

Finally, emergent behaviours is a key feature of ABMs where the space-time processes described at a lower level (micro) of the system lead to complex phenomena at the higher level (macro) of the system that can only be explained in hindsight. In simple terms, simple behavioural rules at the micro level can lead to complex behaviours at the macro level. This phenomenon initially described as swarms in (Bonabeau, Dorigo, & Theraulaz, 1999) can also be observed in the context of the electricity distribution. Indeed, the interactions of the different actors such as consumers, solar panels, and batteries at the premise level can influence the flow of electricity at the zone substation depending on their consumption, the environmental conditions and the battery control algorithms chosen. This characteristic of emergence is one of the key aspects to justify the use of ABM in our research. By describing what happens at the micro level and see what happens at a macro level we might be able to avoid surprises (Hall, 2011) by being more aware of the trajectory the system might take when people are adding more solar panels in some specific areas or when new technologies will start to really have an impact (e.g. with the introduction of electrical vehicles or small-scale batteries). Additionally, this approach can facilitate understanding the impact different policies might have on the whole system (Chappin & Dijkema, 2010). For example, policies to encourage the adoption of some technologies could be understood by seeing where these technologies would be taken up at the grid level and by how much it would alter the electricity flow. In addition, different tariff options could be trialled to see their impact on the infrastructure and over time as some tariffs could encourage the use of battery systems at given times of the day, and see if they are indeed good options to implement.

2.2.4 Agent-based models and multi-agent systems modelling

The terms agent-based models and multi-agent systems are often found together and sometimes used interchangeably; however, they may differ in their purpose. As described in (van Dam, Nikolic, & Lukszo, 2012), an agent-based model is set up in the view of discovering what would happen as the different entities, that are represented and linked to one another, behave. In a way, it is built with the question "*What will happen when* …?". They do not aim at reaching a certain goal, but rather see what the state of the overall system will be as the different entities' actions are played out over time. A multi-agent system on the other hand can be built so that a given desired emergent state of the system is achieved. Similarly to agent-based modelling, the different entities are described along with their interactions, but as the system's emergent behaviour appears, actions will be taken by the different agents to rectify the situation if it does not answer the desired criteria. This is still a bottom-up approach, but the agents cooperate with one another to resolve all conflicts so that the desired system state is observed. The question that can be asked when using multi-agent systems is "*How can I make…?*"

This is a special case of the use of multi-agent, which is used for distributed problem solving (in robotics for example). More generally, in the domain of complex and adaptive systems "*a Multi-Agent System (MAS) contains an environment, objects and agents (the agents being the only ones to act), relations between all the entities, a set of operations that can be performed by the entities and the changes of the universe in time and due to these actions*" (Ferber, 1999) which purpose corresponds to the one of agent-based modelling.

In this research, we are interested in agent-based modelling for their specific properties of seeing what will happen at the system level, as the different agents follow their own path. References throughout this thesis regarding agent-based modelling will be in regards to the definition stated above which is that an agent-based model is set up in the view of discovering what might happen as the different entities that are linked together behave.

2.2.5 Agent-based models in the electricity sector

The use of agent-based modelling in the electricity sector has mainly been applied to analyse market design of power markets for large-scale electricity systems (Bunn & Oliveira, 2007; Chan, Young-Jun, & Macal; Conzelmann, Boyd, Koritarov,

& Veselka, 2005; Gnansounou, Pierre, Quintero, Dong, & Lahlou, 2007; North et al., 2002; Weidlich, 2008). For example, Argonne has developed EMCAS, The Electricity Market Complex Adaptive System (North et al., 2002). This model aims to "*capture and investigate the complex interactions between the physical infrastructures and the economic behavior of market participants that are a trademark of the newly emerging markets*" (Conzelmann et al., 2005); "*EMCAS couples short term simulation for bidding strategy evaluation and asset/portfolio optimisation, with long term analysis capability for generation investment planning*." (Foley et al., 2010). Another framework is NEMSIM, National Electricity Market Simulator, which engine is called Genersys (Batten & Grozev, 2006) which has been developed by CSIRO in Australia. Also built using agent-based modelling, it represents Australia's NEM "*as an evolving system of complex interactions between human behaviour in markets, technical infrastructures and the natural environment*" (Batten & Grozev, 2006). Finally, (Weidlich, 2008) describes an agent-based electricity market simulation model that can be used for methodologically supporting questions of how to best engineer markets in the electricity sector. The model deals with three types of markets: a day-ahead market, a balancing market and a market for exchange for CO_2 emission allowances.

While these frameworks differ in their goals from the one in this study, they show that agent-based modelling has been successfully applied to large-scale systems in the electricity sector. The models presented above not only have agents of different types, representing the market participants in single or multiple markets; they also have the physical infrastructure considered. However, the infrastructure represented is the transmission network, and generators considered in the bidding process are large generators only.

Another area for which ABM has recently been applied to in the electricity sector is in the demand response management area. (Boait et al., 2013) presents an agent-based model that models household electricity consumption and supply where price signals are used to incentivise the user to smooth out their load. This model was done in the view to balance the conflicting interests of the different supply and demand actors as well as the distribution network operators. However, at this stage of their work, the authors have not implemented the physical elements of the distribution network.

Infrastructure transitions of the electricity sector has also been assessed using agent-based models (Chappin & Dijkema, 2010). For this, the authors have developed a typology leading to the assessment of different model designs of possible futures. Models of the impact of CO_2 policy on the power production sector, of the transition to global liquefied natural gas infrastructure and the introduction of LED lighting system have been assessed using ABM. However, again, the analysis only describes the physical infrastructure at the generation level.

One tool however that models the distribution network using ABMs is GridLab-D. GridLab-D is a power system modelling and simulation environment that has been developed in the U.S. over the last decade (Chassin, Schneider, & Gerkensmeyer, 2008). The system describes all the elements that compose the distribution grid up to the transmission grid, taking into account the technical characteristics of the system, that is, describing the physical components as well as their behavioural interaction. Similarly to the system described in this research, it aims at seeing the impact of a change in one part of the system to another one and has the different aspects of the system tied together. It considers a few areas of the electricity sector, such as the electricity market: "*It's a power systems simulator, market simulator, communication simulator and building simulator, all tied into one. Every piece shares information with every other piece to build a clearer picture of how the electric grid will evolve over time*."(2012). GridLab-D uses different levels of detail and combines different kinds of modelling techniques depending on the need of the calculations. The state of millions of independent devices is calculated locally using differential equations – this is used, for example, for calculations such as the energy transfer through a premise's wall; power flow analyses can be performed using gauss-seidel algorithm and finally some models such as the end-use models for appliances are implemented using agent-based simulation methods; however, the extent to which type of agent-based simulation method is used is unclear.

GridLab-D and the M&S application presented in this thesis have similar goals which consist in having a detailed representation of the distribution grid where the different elements and their impact on one another can be taken into account. However, one of the differences between the two simulation environments which is of interest in this thesis lies in the methods employed to build the software system. In a description of GridLab-D (Chassin, Fuller, & Djilali, 2014), modules are mentioned that allow bringing different analyses together - these include AC power flow, building thermal and retail electricity modules. These modules differentiate the domains of analysis, and probably interact through interfaces, although this is not clear. In this research, we aim at going one step further, where within a sub-domain, a compositional approach will be taken to describe the different agents which components will then be brought together to form the agent-based model.

2.3 BUILDING AGENT-BASED MODELLING AND SIMULATION APPLICATIONS

This section describes the methods and tools that can be used to build agent-based M&S applications and what can be considered once an ABM M&S application becomes large.

2.3.1 Developing agent-based models

As described in the previous section, agent-based models are made of agents that are self-contained, self-directed and autonomous, amongst other characteristics. They can sometimes be seen as objects that are connected to one another with the added characteristic that an action is associated to them that will define the way they respond to a stimulus or their environment. Taking this point of view, methods from the object-oriented paradigm, such as UML (Unified Modelling Language), can then be used to develop the conceptual model. UML is a popular specification that is used to model not only application structure, behaviour, and architecture, but also business process and data structure (Booch, Rumbaugh, & Jacobson, 2005). This method is therefore very useful as it encompasses not only the object definition but also the behaviours which are defining the agent; it can therefore not only be used to support agent-based models in the design phase but also in the communication, and is consequently commonly used to support developing agent-based models (Macal & North, 2006).

Other methods describing agent-based models are also available but they are not necessarily used to build the model but rather to communicate it, e.g. for publishing, so that the model can be reproduced. Such methods include code templates, like those used by (Railsback, Lytinen, & Jackson) and text-based methods such as the ODD (Overview, Design concepts, and Design Details) protocol (Grimm et al., 2006). These allow other modellers to reproduce the agent-based model that has been described in the literature.

2.3.2 Implementing agent-based models - ABM toolkits

One of the most straight-forward ways of implementing an agent-based model is to use one of the many toolkits that are available. With the growing popularity of agent-based models over the years, many tools have been built that offer very diverse functionalities depending on the purpose and users for which they were developed. Assessment of these toolkits are presented in the literature (Berryman, 2008; Najlis, Janssen, & Parker, 2001; Nikolai & Madey, 2009; Railsback, Lytinen, & Jackson, 2006), using different evaluation criteria. Such criteria are, to name a few, the tool's programming language, the type of license (proprietary, free license (open source, BSD, GPL…), associated third party licenses…), the operating system (Windows, Linux, Macintosh), the domain of application (general purpose, ecological modelling, multi-agent simulations…), or the type of user support (API, Consulting, Tutorials, FAQ, Wiki…).

Amongst the many available ABM toolkits, three of the most popular ones were experimented with, in the context of this research: Repast Simphony (Argonne National Laboratory, 2011), MASON (Luke, Cioffi-Revilla, Panait, Sullivan, & Balan, 2005) and Agent Modelling Platform (AMP) (The Eclipse Foundation, 2011). While these software platforms offer good support to building ABMs, as the model grows, it can become difficult to expand and maintain the model, and a decrease in speed during the simulations can be observed as the number of agents increases. Because of this, some of these tools have been further developed to support parallel and distributed ABMs, as the need for faster execution times and support of larger and larger models has arisen. This is the case for D-MASON (Cordasco et al., 2013), which is the parallel and distributed version of MASON, or Repast-HPC (High Performance Computing) (Collier, 2013) for Repast Simphony.

JADE (Bellifemine, Caire, Poggi, & Rimassa, 2003) which is a middleware for distributed multi-agent application based on peer-to-peer communication architecture has also been used to build ABM M&S applications. It has become a very popular application in the domain of electrical engineering and has been utilized in many applications for the electricity grid (Abras, Kieny, Ploix, & Wurtz, 2013; Jun, Junfeng, Jie, & Ngan, 2011; Ren, Zhang, & Sutanto, 2013) as it is FIPA-compliant, and suits well the requirements when modelling the electricity grid – components distributed over space, requiring two-way communications. JADE is interesting to use especially when building distributed applications where agents can be implemented on different devices. The interest here lies in the potential of the deployment of the code over functional units on the distribution grid. As such, it can be assumed that code implemented for an agent in an ABM might be used by an agent that is then deployed on the micro-grid for example.

While these parallel and distributed implementations allow building larger agent-based models, they do not necessarily support well building the model in a flexible and extensible manner. Indeed, the model still needs to be built in a central place, even if the agents can be run on distributed cores. Also, reuse of parts of these toolkits is not always feasible and when it is, it might not always support well the implementation of a modular agent-based model. As an example, MASON was initially selected as the ABM engine for the software platform presented in this thesis. It was selected for its ease in separating the engine (which was of interest) from the rest of the platform as well as for its execution speed. However, it was later replaced by our own implementation of two schedulers in Java (a sequential and a parallel one), as this allowed having the full implemented model consistent with the rest of the ABM software environment, built with extensibility and flexibility in mind.

Following this, understanding better what large-scale ABMs requirements are is essential. From (North & Macal, 2007), chapter 10, "*large-scale ABMS is usually done with computer-based ABMS tools. The criteria for selecting a good tool for a specific large-scale modelling task should include consideration of each system's time scheduler, communications mechanisms, interaction topologies, allowed architectural choices, capabilities for storing and displaying agent states, large-scale development support, and special topic support*." While these criteria are for the selection of a good tool, they still apply when building our own software environment that will hold a large-scale ABM, and are of great importance.

2.3.3 Building large-scale agent-based M&S applications

Often models evolve organically from small to rather large sizes as more information becomes available. This evolution might not be planned, either because the information is not initially available to have a very detailed model, or because the requirements of the simulations have suddenly increased. When this happens, it is possible to upgrade the agent-based M&S application, and some protocols can be followed to ease this upsizing process. For example, (Parry, 2012) presents a protocol that enables to go from a small simulation to a larger one, which includes the questions that need to be asked in order to scale up more easily, as well as the different technical options that can be taken. These are given as**:**

- "Optimize existing code.
- Clearly identify scaling requirements (both for now and in the future).
- Consider simple solutions first (e. g. a hardware upgrade).
- Consider more challenging solutions.
- Evaluate the suitability of the chosen scaling solution on a simplified version of the model before implementing on the full model."

These considerations can be useful when a simulation needs to be performed and not enough resources are available to refactor the whole M&S application for example, but they seem more like a Band-Aid solution rather than a well-thought implementation process. Planning in advance to build a large-scale ABM following well-thought processes would be much more efficient rather than fixing the scale problem as an after-thought.

Challenges of growing models or software applications are well-known in the software engineering domain, and the methods developed to deal with the development of large software can be transferred to this context, to inspire the process as well as the requirements when building large-scale ABMs. The following section describes methods used in the software engineering environment which could be transferred to the development of large-scale ABMS.

2.4 APPLYING METHODS FROM THE SOFTWARE ENGINEERING DOMAIN TO THE DEVELOPMENT OF LARGE-SCALE AGENT-BASED MODELS

Developing a large-scale agent-based M&S application can quickly become very complex. Having an appropriate software environment and adequate development methods is therefore important. This section presents methods, software architectures and available techniques used when developing large software environments. Looking at these three aspects can inform the way the agent-based model and the simulations can be built.

2.4.1 Software development processes

Two main software development methods can be identified in the literature: the Waterfall method (Royce, 1987) and the Agile Software Development method (Dingsøyr, Dybå, & Moe, 2010; Thomas Stober & Hansmann, 2010).

The waterfall method is a sequence of processes that describe the different phases of a software system development: requirements, design, implementation, verification and maintenance. It was first formally described by Winston W. Royce in (Royce, 1987) although not coined at the time. This method was initially adopted from the manufacturing industry where all the different processes are described at each phase. It is only possible to move to the next phase when the current one is complete as modifications at a later stage are often cost-prohibitive. The waterfall method has the advantage that because the project requirements are thoroughly defined, it gives a good idea of the project cost and timeline, as well as what the program will be like in the end. Also, its emphasis on meticulous documentation allows for improvement in the future existing program and makes it easier for a new team member to come to speed with the code. However, because it relies heavily on initial requirements, if they are found faulty or a change needs to be made, the project needs to start again which can lead to great delays in the delivery of the final product. Because initial requirements are often difficult to precisely define, this method, while still in practice in some software development groups, has often been superseded by the Agile Software Development method.

The Agile Software Development method is an iterative and incremental method which suits environments that are evolving, focussing on customer value and promoting participation and collaboration (Thomas Stober & Hansmann, 2010), as defined in the Manifesto for Agile Software Development (Beck et al., 2001). Because software systems changes are not as costly as manufacturing one, it can be more beneficial to start the implementation at an earlier stage while the software requirements may not be definite. As David Parnas, puts it in (Parnas & Clements, 1986), “*Many of the [system's] details only become known to us as we progress in the [system's] implementation. Some of the things that we learn invalidate our design and we must backtrack*.” As a project advances in its execution and the software evolves in its implementation, many changes might need to occur. The Agile method has the advantage that it allows changes after the initial planning, and adapts to evolving client's needs, which facilitates the addition of new features. Also, client's feedback at the end of a sprint can be taken into account, ensuring the software is evolving in the direction the client wants. However, one of the disadvantages of this method is that a series of features can be implemented resulting in a product that has not been well thought.

Many methods and techniques constitute the Agile methods amongst which are scrum, lean software development, test-driven development and Extreme Programing (XP). One of these methods, Extreme Programing (Beck & Andres, 2005) was developed to improve the quality of software and the responsiveness to changing customer requirements. For this, a cycle or feedback loop methodology is used, with different steps such as: pair programing, unit testing, pair negotiation, stand-up meetings, acceptance of tests, interaction plan and release plan. This is a very interesting method that has proven to be efficient, and is used in many private software development companies.

Both these methods have advantages and inconveniences, and could be applied to building our ABM M&S application. However, because of the evolving nature of the electricity grid in terms of technologies, as well as the way they are used, integrating this new knowledge in the model and its implementation will be better supported by following the Agile method. Similarly, when defining the simulation specifications, two approaches based on these methods philosophy can be taken. Taking the waterfall model approach, the simulation would be built after all the requirements are known, while in the case of the Agile method, the simulation could be built in an iterative manner, where the different elements could be added and/or refined as more information becomes available. Again, following the Agile method would be most appropriate.

In conclusion, building the model and the simulation so that it can evolve over time as more information becomes available requires software engineering processes that support flexibility and extensibility, which the Agile method supports well.

2.4.2 From the separation of concern to modularity

Following this example of transferring the practices of software engineering to the process of building an M&S application, the structure of the ABM implementation can also be inspired by known architecture from the software engineering domain.

As such, a programming paradigm of interest in the context of this research is the modular programing one. Contrary to a monolithic approach where everything is defined in one single support element, modular programming is based on separating the different functionalities of a program into modules that are independent of one another and that can be interchangeable. Modules then follow the principle of separation of concern (Dijkstra, 1982) and ensure that the logical boundaries between components are respected. These modules have their own information hidden, which is what Parnas defines as modularity (Parnas, 1972) and they can be integrated into the software structure using interfaces that define the nature of the exchange of information for the modules to work together.

In (Baldwin & Clark, 2000), chapter 3, the concept of modularity is defined using two ideas: *"interdependence within and independence across modules",* and *"abstraction, information hiding and interface". By "interdependence within and independence across modules",* the author means that the elements described within a module will be strongly connected to one another; interconnectedness will be explicitly described. Within a module, the elements are abstracted and the information is maintained within, hidden from the rest of the system. The links of the elements between modules will be weak, and enabled by the interface only. The interface will provide the boundaries of the modules so that that they can come together to form the overall system. Having independence across modules allows them to be replaced while keeping the overall functionality of the system. The replacement can be either functionally identical but implemented differently, or the replacement could be extending the functionality. While the definition of modules and their use within this reference is not specific to software engineering, they can largely be adapted to it.

Several benefits in using modular designs can be identified such as allowing parallelism in design and testing, increasing the number of design options by mixing and combining modules, and adapting locally within hidden modules (Baldwin & Clark, 2000).

2.4.3 Modular approach to software implementation

Many programming languages have been developed with the concept of modularity in mind, with object-oriented programming or agent-based programming amongst them. While modularity can be seen at the object or agent level, which is a very fine level of modularity, it can also happen at a higher level of granularity. Component-based software engineering (Szyperski, 1997) is an example of a modular implementation at the software level. This approach has been used in supporting building simulations, such as discrete-event simulations as presented in (Buss, 2000). In their paper, the authors define a component as a "*monolithic programming entity whose external interface consists only of property accessor/mutator methods, of action methods, and event handler methods*"

In this thesis, modularity needs to be at the software level, i.e. at the component level, as well as at a fine level through the use of agents that are defined in an agent-based model. Some work has been undertaken in the area of component-based approach for multi-agent systems. In (Meurisse & Peschanski, 2007) the authors believe that the component concept and technology may help in the construction of multi-agent systems, as we do in this thesis. They mention composability at two levels: the system level, which may consider each agent as a component and at the agent level, where the internal architecture is built in a modular manner for structuration, (de)composition and reuse. They have developed MALEVA where each agent’s behaviour can be composed using another one. For this, two ports are used, one data port and a control port that allow the communication and action of the agents at runtime. The approach undertaken in this thesis differs from the one undertaken for MALEVA as the main focus is using a component-based approach at the system level as well as at the agent level. Additionally, MALEVA uses connectors and has output interfaces so that the output of one behaviour is the input of another one to form a chain of complex behaviours. While this approach could be used in this thesis as well, our use of composability is rather more oriented in having alternative or complementary behaviours that can be chosen to compose the overall agent's behaviour. Further, composability will not be used for behaviours only, but for describing the static characteristics of the agents, as well as to form groups of agents of same type that can be brought together to form the agent-based model.

2.4.4 OSGi for writing modular software

OSGi (formerly Open Services Gateway Initiative) (OSGi Alliance, 2013) is a specification that enables writing software in a modular manner. The modules in OSGi are called bundles, which have the properties of being installed, started, stopped, un-installed at run-time. This means that applications using OSGi result in being very modular and dynamic, where a bundle can be modified without the need for the system to be rebooted. These features are of particular interest when building multi-agents systems, which can have the different agents implemented within a bundle installed on a device, but also in the simulation phase when developing the system. JADE has been paired with OSGi (Quarantotto & Caire, 2010), enabling building more dynamic applications. Its use has been illustrated in (Carneiro, Novais, Costa, & Neves, 2010) showing examples of intelligent environments developments, with the example of virtual environemnt for dispute resolution, and another one for the monitoring of elderly in their homes.

This specification has also been used by the Eclipse Community, whose plugins are OSGi bundles (Vogel, 2012) and are the smallest units of modularisation. One important feature of plugins is their capacity to easily add or remove a feature from the simulation as it runs as well as when setting up a simulation. This can be leveraged from, especially as the simulation can be built in a flexible and extensible manner and will be considered when developing the software that supports building our compositional agent-based model.

2.4.5 Composability for building simulations

Modularity happens at the model implementation level, to provide a flexible and extensible model, which then facilitates building many simulations by simply combining these modules together. This process of bringing the modules together is called composability, defined by (Petty & Weisel, 2003) as "*the capability to select* *and assemble simulation components in various combinations into valid simulation systems to satisfy specific user requirements.*". By having a modular architecture, rapid set-up of simulations, where different configurations can quickly and easily be tried, is possible.

Composability to agent-based simulations has been achieved where existing simulators are reused and combined to generate a simulation (Benali & Ben Saoud, 2011). Another example of composability lies in the domain of smart-grid (Schutte & Sonnenschein, 2012) where the authors have created a test-bed for Smart Grid control mechanisms (MOSAIK), that reuses existing simulators and brings them together to create large-scale scenarios. This thesis does not aim at reusing existing simulation models from independently developed simulators, however, previously developed modules within the framework need to be reused and brought together to create the simulations. Composability is therefore an important concept when developing a modular agent-based model and simulation application. The modules that are going to be developed will need to have a mechanism allowing connection between them (often called a port where the input and the outputs match) as well as have transitive properties (if *a* is related to *b*, and *b* is related to *c*, then *a* is related to *c*). Further, because of the interdependency due to the agent nature of the modelling technique, additional properties to link the different agents developed in independent modules will need to be defined.

Another important point in building the simulations is the process with which these are developed, considering that the number of possible simulations will grow as the model itself grows. The process to model socio-technical systems has been described in (Nikolic & Dijkema, 2010) along with the requirements to model them. They mention that the process in building a complex adaptive system needs to be itself complex and evolving. They have developed an evolutionary modeling process, which consists in repeating a series of steps that continually generate hypotheses and conclusion for the social process and the technical development. These different steps include: the technology (the modelling and simulation paltform), the social process (including the stakeholders to get to an hypothesis), the domain knowledge (the domain representing the system, e.g. electrical engineering and social science for behaviour) and the facts (that can be the physical infrastructure, e.g. how the transformers function as well as a PV). These four aspects will be considered during this thesis, with particular emphasis to the technology that supports such process.

2.4.6 Compositional design and agent-based approach

The characteristics that motivate a compositional design and which that motivate an agent-based approach can be summarised as follows:

- The ABM approach is motivated by:
  - Assessing information at the individual level that result from their interaction with the environment or other agents within the system;
  - Observing emergent properties of the system resulting from individuals' actions and interactions.
- The modular design is motivated by:
  - Rapid prototyping of new agents (assets and/or behaviours); and facilitation of the extension of the ABM;
  - Rapid setup of simulations by putting the different components together to represent the system and its functioning.

In a way the agent-based modelling approach is to ensure that the different actors impacting the system are taken into account, and that through their evolution, emergent properties of the system will be observed. The component-based approach ensures that large-scale agent-based models and simulations can be developed in a flexible way.

As this method is applied to the distribution electricity network, it can be illustrated as shown in Figure 2-7. A representation of the distribution network is given on the left-hand side where all the different entities are represented capturing the information modelled in the agent-based model where every single element and their connection to other agents is represented. The implementation of these agents however needs to be done in a modular manner as represented on the right-hand side of Figure 2-7. The different agents defined within modules of functionalities can then all come together as a simulation is being composed.

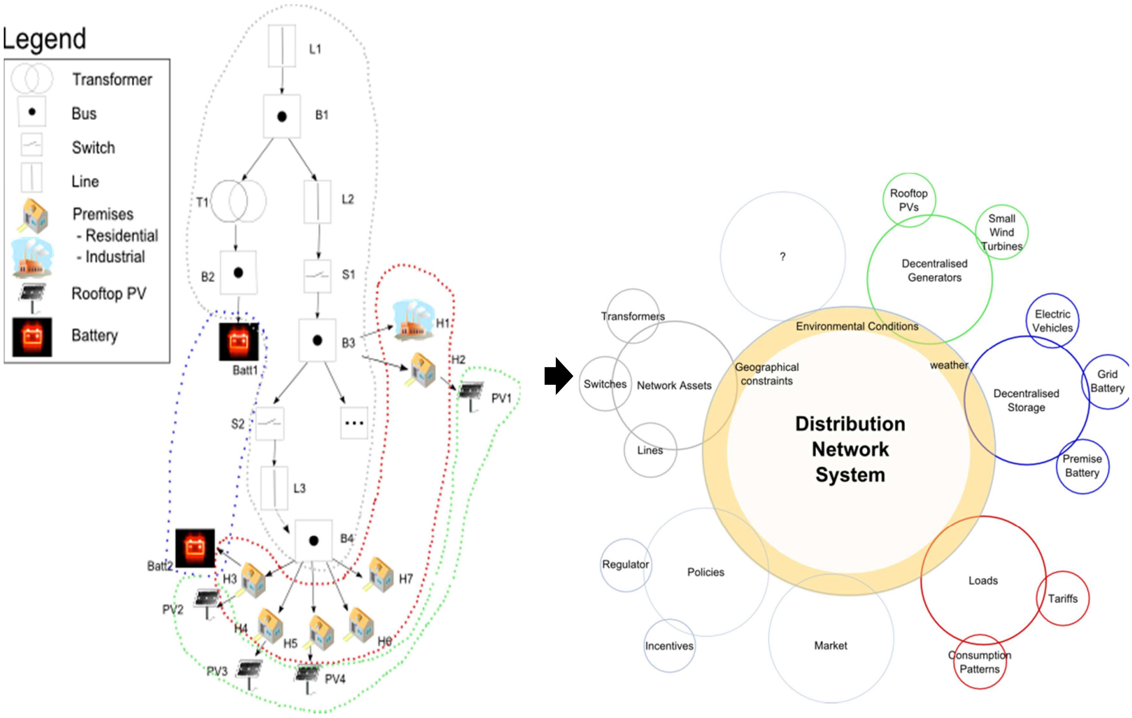


Figure 2-7 - From the system to its description, represented as fine units of modularisation.

2.5 SUMMARY

A gap exists in terms of analysis tools and models when scaling up the studies of local electricity generation to the distribution level. Agent-based modelling which is a bottom-up modelling technique can fill this gap by accounting for every single unit of generation or consumption of electricity at the lowest level of the system (e.g. premises, rooftop solar PV) and seeing how the load changes at higher levels in the system (e.g. transformers in the distribution network). However, because the electricity sector is a very large complex system, as the model grows, it can become very difficult to maintain and expand it, as well as build many simulations to assess possible trajectories of its evolution. A modular or compositional approach, widely used in the domain of software engineering can be used to support building large-scale ABMs which is strongly encouraged by some authors of ABMs (Nikolic & Dijkema, 2010). However, no technical solution as to how to develop such a modular ABM is provided and no agent-based model has been found in the literature that uses such approach. Software environments have been developed to deal with large-scale ABMS and some protocols are available to help move the simulations to larger scales; however, nothing has been found that guides the development of large-scales ABMs from their conception, neither provides rapid development of model extension and setup of simulations, and none using a compositional approach.

In (Hamill, 2010), Hamill sets out a vision of having a library of building blocks for agent-based models. These building blocks would capture a specific environment, or agent, or group of agents and by bringing them together a modeller would be able to set up a simulation more easily, which would be especially interesting for the non-programmer.

This is what this thesis aims at doing, by applying the principles of modularity and composability to the development of ABM M&S applications of electricity distribution networks subject to technological change.

Chapter 3: A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network

In this chapter, which has been written as a conference article[1], we present a planning tool that was built to find the optimal investment strategy for distribution networks over large areas and long planning horizons. The physical configuration of the infrastructure describing the network elements and their functioning at a fine level of detail, both geographically and temporally are described in this tool. The two modelling techniques that are used are introduced: agent-based modelling (ABM) and particle swarm optimization (PSO). This tool was developed so that it could support the examination of the system as a whole, where the impact of a change in one part of a system on other parts can be assessed. A case study of Townsville, Australia, is used to illustrate the platform implementation and the outputs of a simulation.

It has to be noted here that while PSO is mentioned in this chapter, it is not part of this thesis which focuses solely on agent-based modelling and simulation applications. However, because this research is within a broader context of research which includes the use of PSO, it is briefly presented in this chapter.

[1] The article was published in the conference proceedings of the 2012 Winter Simulation Conference, in Berlin.

A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network

Fanny Boulaire, Mark Utting, Robin Drogemuller, Gerard Ledwich, Iman Ziari
Queensland University of Technology, 2 George Street, Brisbane, Queensland 4000, Australia

A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network

ABSTRACT

With an increasing number of small-scale renewable generator installations, distribution network planners are faced with new technical challenges (intermittent load flows, network imbalances…). Then again, these decentralized generators (DGs) present opportunities regarding savings on network infrastructure if installed at strategic locations. How can we consider both of these aspects when building decision tools for planning future distribution networks? This paper presents a simulation framework which combines two modelling techniques: agent-based modelling (ABM) and particle swarm optimization (PSO). ABM is used to represent the different system units of the network accurately and dynamically, simulating over short time-periods. PSO is then used to find the most economical configuration of DGs over longer periods of time. The infrastructure of the framework is introduced, presenting the two modelling techniques and their integration. A case study of Townsville, Australia, is then used to illustrate the platform implementation and the outputs of a simulation.

3.1 INTRODUCTION

Following commitments by Australia to reduce its greenhouse gas emissions, the target of generating 20% of energy using renewables by 2020 has been set (Australian Government, 2011b). To help reach this goal, measures have been put into place by the government to encourage individuals to invest in small-scale generation units, such as rooftop solar panels (Queensland Government - Office of Clean Energy, 2011) and (Australian Government - Clean Energy Regulator, 2012b). These decentralized generators (DGs) have advantages not only in terms of CO_2 emission reductions, but they can also support the peak load growth if installed at strategic locations on the network, potentially saving money on infrastructure investments. Indeed, Australia's peak electricity consumption, which is the metric used when designing electricity networks (generation and infrastructure), has rapidly increased over the last few years. For example, Queensland has observed a 7% increase in peak demand on average over the last 7 years (Queensland Government, 2009) with up to 14% increase for some years. Knowing that approximately 10% of the electricity distribution network capacity is built to meet the peak demand, which only occurs for approximately 1% of the time (Queensland Government, 2009), understanding how DGs can best be used to shave peak demand is crucial when planning for infrastructure maintenance and extension.

However, despite these opportunities, DGs are causing new technical challenges to distribution companies, which have to manage intermittent load flows, network imbalances… preventing the installation of DGs in some areas that have reached a high percentage of penetration (Hall, 2011). While some countries in Europe have managed to reach quite high percentages of variable renewable generation (over 10% for wind for example), this was achieved thanks to cross-border transmission links (Kirby & Milligan, 2008). Australia, being isolated from other countries, needs to deal with the technical challenges in a different manner. Also, because of the lack of experience with high penetration rates of variable electricity resources worldwide, very little is known about what will happen when these generation types increase to very large proportions (Komor, 2009). Consequently, distribution companies are looking at understanding such effects using modelling techniques which can investigate different scenarios for planning future energy grids.

The project described in this paper aims at assessing the impact of the large-scale introduction of renewables on a distribution network, managed by Ergon Energy, one of the two electricity distribution companies that serve the state of Queensland, Australia. Due to the network spare nature (it covers 1.7 million square kilometres and provides power to approximately 680,000 homes and businesses across regional and rural Queensland) and its vast number of assets (approximately 150,000 kilometres of power lines, 1 millions of poles) many components need to be taken into account, in addition to the introduction of renewable generators, when planning for the future network. A question then arises: “How can we best manage the network that will allow for a mix of network extension and integration of distributed generation that will optimize the overall system in terms of cost benefits and demand?”

To answer this problem a framework has been developed. This framework supports the examination of the system as a whole, so that the impact of a change in one part of a system on other parts can be assessed. Contrary to the current practice where critical areas are identified and assessed using static analyses to assess the level of capacity required to service an area (Ergon Energy, 2010), the framework considers the system as a whole, whether or not the areas have reached capacity. This means that all the components making up the network need to be represented and the evolution of the system needs to be captured over-time. As such, a dynamic analysis of the distribution system is required that can assess both the consumption (for end-user premises) and the production (of renewable DGs) over time and at specific locations. This dynamic assessment can then be used in the analysis to find the most economical configuration of DGs for long-term planning.

The framework presented in this paper consequently performs analyses over different time scales and calls on different techniques. Two modelling techniques are used: agent-based modelling (ABM) and particle swarm optimization (PSO). ABM is used to represent the different system units accurately and dynamically, following the changes over time and at different levels of detail in the distribution network. Load duration curves which are output from the ABM simulation are then used in the PSO module to find the most economical mix of network extension and integration of distributed generation over long periods of time. Combining these two modelling techniques allows taking advantage of each method’s strength to obtain the necessary simulations over different timeframes. Another key feature of this framework is its ability to model both (1) the technical network constraints and (2) the economic and sustainability challenges of minimizing cost and carbon intensity.

This paper presents in the first instance the overall architecture of the framework, introducing the two modelling techniques used in the framework's components and their interactions. The implementation of the platform is then illustrated using a case study of Townsville, Australia and results of the discrete-event simulation are discussed.

3.2 AGENT-BASED MODELING AND PARTICLE SWARM OPTIMIZATION

Agent-based modelling is a technique used to model complex systems comprised of autonomous and interacting agents. An agent-based model (ABM) can be described as a "class of computational models for simulating the actions and interactions of autonomous agents (both individual and collective entities such as organizations or groups) with a view to assessing their effects on the system as a whole" (Wikipedia, 2011). This modelling technique is becoming increasingly widespread with applications in domains as diversified as consumer behaviour modelling, traffic congestion analysis and electricity market simulations. One of the reasons for such an uptake of this technique is that we live in an increasingly complex world, but also that the computational power currently available allows computing "large-scale micro-simulation models that would not have been plausible just a couple of years ago" (Macal & North, 2006).

As its name indicates, agent-based models are composed of agents. While there is no universal agreement on the definition of an agent, most modellers share the same view regarding the properties of an agent. Agents have the following attributes (Macal & North, 2005): agents are autonomous and self-directed, they are modular or self-contained, and they are social, interacting with other agents. They can also have additional properties such that: they live in an environment, they have explicit goals that drive their behaviour, and they learn and adapt their behaviours based on their experiences.

These agents can be specified at various scales, defining the granularity of the model. As they are autonomous, decision-making heuristics, learning rules or adaptive processes have to be included in the model as well as an interaction topology. Such properties make using ABM for the electricity network adequate. Indeed, the components making up the network can be defined as agents, with the network topology captured through the agents' interactions. The behaviour of the agents can be defined independently reflecting the way that end-users consume electricity according to their needs and the activities they undertake. Through building the system from the lower level (e.g. where an agent can represent an asset on the network such as a transformer) to a higher level (e.g. a substation), an emergent phenomenon can be observed and the system can be assessed as a whole. Additionally, the encapsulation of algorithms and data within each agent improves flexibility when configuring alternative scenarios.

While agent's behaviours can be optimized, a clear distinction has to be made between agent-based modelling optimization and full optimization of the system, as it is fundamentally different in nature and purpose. The agents' intention is not to optimize the system as a whole; they typically try to optimize only their own behaviour. This could be regarded as doing local optimization over the system, at the agent level, which might result in a very non-optimal system. While ABM is not an appropriate technique for system optimization, it can be combined with optimization methods to bring a compromise to the full analysis, as implemented in this framework.

Optimization was used as the technique to plan for the most economical mix of renewables to be installed on the distribution network. Because optimal planning of distribution networks is a large-scale mixed-integer and nonlinear problem, classified as an NP-hard problem, conventional optimization methods (e.g. linear programing, integer programing) are not good candidates for such problem. However, PSO, which is a population-based and self-adaptive optimization technique, is known to effectively solve these types of problems, and was employed here. PSO has been effectively applied to different optimization problems in power systems, one of which is the generation expansion problem which "consists in determining what, when, where and how to install new generation units in order to meet the power system requirements while constraints regarding load demand, reliability, operating conditions, power quality, and security are met" (del Valle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley). This problem is formulated so that the least cost of the investment as expressed in the objective function can be found. Such problem is usually applied for the placement of high-voltage generation units, but is adequate to the case of low-voltage generators.

PSO, which is a stochastic-based method, handles a population of individuals in parallel, to probe areas where the optimal solution is located. The individuals are called particles and the population is called a swarm. This optimization method leads particles in a swarm toward the optimal point using an adaptive velocity. Many different variants of the basic PSO can be used (del Valle et al.) the one chosen for this problem is a hybrid of evolutionary programing and PSO, as it uses mutation and cross-over when updating the position of the particle, as described in (Ziari, Ledwich, Ghosh, Cornforth, & Wishart, 2010).

3.3 RELATED WORK

A large number of tools are available for various analyses in the electricity sector. For example, in their review (Foley et al., 2010) have described 7 models applied to electricity systems; in (Connolly et al., 2010) 68 tools have been identified for analysing the integration of renewable into various energy systems, 37 of which are described in more detail; and many more tools are available that can be found on the internet, for example (Powertech Labs Inc, 2012; ROAM Consulting, 2012), without counting the ones developed internally by the different electricity authorities. In many of these tools, analyses of the feasibility in installing renewables at given locations, calculating how much they are expected to produce at different times of the year, whether it be small-scale or large-scale renewables, is possible. These types of tools aim at finding the most profitable location to install large-scale generators, and therefore detailed analyses are available. However, no transition in models from site assessment to state-wide assessment of impact of renewables was found, especially when it comes to linking the electricity generation from the renewable systems to the technical constraints of the distribution network.

A framework, MOSAIK (Schutte, Scherfke, & Sonnenschein, 2012) is currently being developed that aims at linking existing simulation models automatically. For this, the technology regarding model interfacing and data flow management is being developed. While this work aims at putting existing models together, it does not address the problem of putting optimization together with models, especially when dealing with linking data at different levels of granularity and over different time periods. The work presented in this paper is innovative in the way that it links planning methods (through the PSO module) taking into account the variability in the network behaviour (through the ABM module) which is usually only considered when performing analyses at the operational level. The data used for the optimization module is obtained after manipulation and aggregation of the data output by the ABM module and the optimization is done over longer time horizons than the ABM. Another framework (GridLAB-D) that aims at providing similar functions as the one proposed in this paper (Chassin et al., 2008) is also currently being developed. The GridLAB-D framework though, only has a few of its planned modules currently implemented, and none deal with the optimization of the placement of DGs.

The work presented here offers an innovative way of combining models at different levels of granularity (different geographical and time scales) to optimize the investments of the distribution network. Network augmentation and introduction of DGs are considered in the optimization part of the simulation informed by the behaviour of the different entities while in operation, allowing capturing the actual variability in the assets performance and not relying on averages only. Additionally, this framework models both (1) the technical network constraints and (2) the economic and sustainability challenges of minimizing cost and carbon intensity.

3.4 MODULAR FRAMEWORK ARCHITECTURE

For this project, not only does the simulation require the implementation of two modelling techniques but it also requires a multi-level representation. The simulation needs to happen over two time horizons (short-term management and long-term planning) and needs to be location specific (where load patterns evolve depending on the area of consumption and generation). Figure 3-1 gives an overview of the framework, showing how the different components represent the systems entities and their relationships, as well as the use of the two modelling techniques.

As can be seen, the framework has different levels of complexity; the simulation is done over varying levels of granularity in terms of agents and time.

Agents in this context can be individual network assets, but they can also be represented as groups of agents whose characteristics result from the aggregation of finer-grain agents. As an example, an agent can be a single photovoltaic unit on a house at the finer level of granularity, or at a coarser level of granularity, it can represent the photovoltaic characteristics of a suburb in terms of penetration rate and generation capacity.

The time scale for this framework contains three levels of detail: ½ hourly, yearly and five-yearly. Simulation runs are performed on a ½-hourly basis for the calculation of the electricity demand at a specific location and for every day of the year; the calculation is such that the overall demand to the grid consists of the electricity required by a household, for example, minus the electricity generated by the photovoltaic unit, if installed, whose output depends on the solar availability. These ½-hourly calculations are then aggregated and yearly simulations can be performed that consider growth in demand due to factors such as population growth or increase in PV penetration. These calculations are done in the ABM module, where the network and its characteristics are defined. The outputs of these calculations are manipulated, so that they can be used for the 5-yearly simulation, which corresponds to the time cycle used when developing network improvement plans. A set of load duration curves is used as an input to a load flow analysis which is iteratively used with the PSO program. Different configurations of the network are assessed and the output results in selecting one configuration for which costs (capital, maintenance and depreciation) of the network have been minimized under the network constraints.

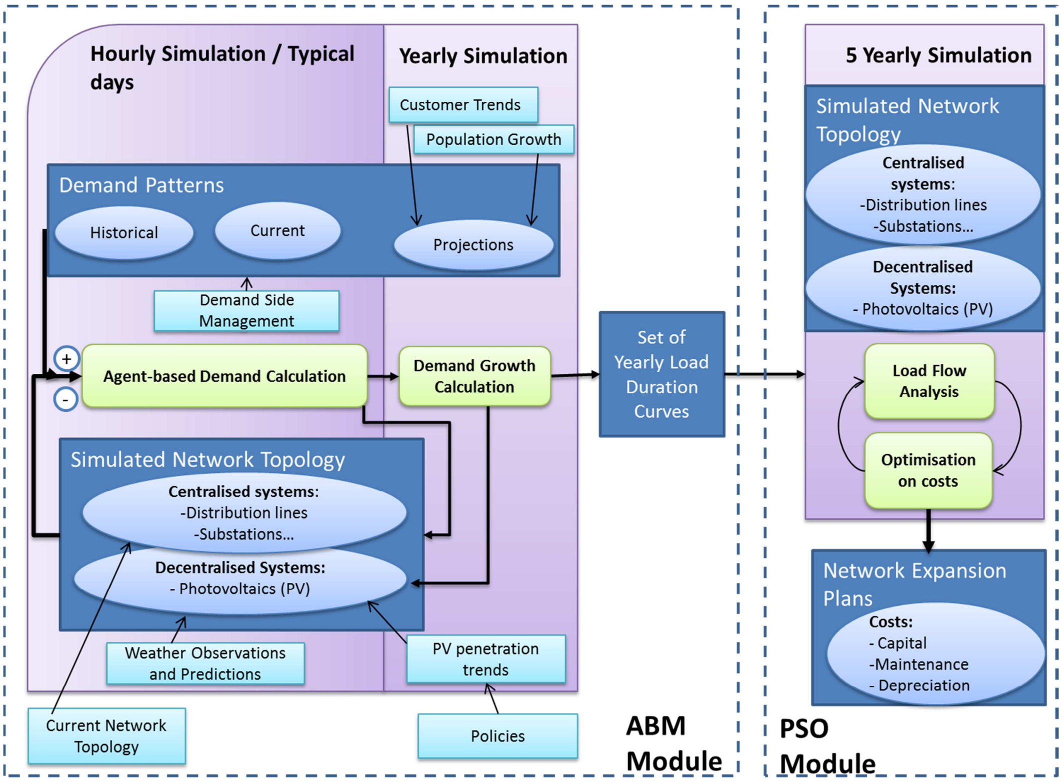


Figure 3-1 - Schematic representation of the ABM and PSO modules and their relationships to one another. The left panel represents the short-term management module for which ABM is used. The right panel contains the long-term planning module which uses PSO. The ABM module performs simulations over 2 timeframes (hourly and yearly) using data describing the network topology and demand patterns. The output of a yearly simulation in the ABM module consists of a set of yearly load duration curves, which become input to the PSO module. The optimization module outputs network expansion plans for which costs have been minimized.

The two modelling techniques (ABM and PSO) are used distinctly over the two modules and each addresses different requirements of the analysis cycle; they can also be run independently. However, by combining them, this framework integrates different elements of analytical complexity to represent the distribution system in a realistic manner, using each method's strength.

3.5 IMPLEMENTATION OF THE FRAMEWORK

The framework described above was then implemented. The following aspects were considered that refined the way of its implementation:

- **The data**: data about the network assets and the load patterns were made available by the distribution company.

- **The modelling approaches**: as mentioned above, agent-based modelling and particle swarm optimization were selected as the techniques employed to solve the problem, which use was tailored to the available data and the need of the problem.
- **The toolkits and development environment**: these were selected according to the requirements set by the problem definition, the modelling approaches selected and the data available.

Details of these three aspects are given in this section. The implementation of the platform is finally illustrated, using a case study of Townsville, Australia, for which the output of a simulation is presented, and the results are discussed.

3.5.1 Description of the data used within the framework

Different data sources were made available for this project by Ergon Energy, the distribution company who partnered on this project. The two main datasets contained information about:

- The **network connectivity**. Information about the different assets, their characteristics and their configuration within the network were available.
- The **electricity demand** at different points within the network. Three types of demand data were available:
  - Quarterly billing consumption data for every premise in Townsville for year 2010
  - Half-hourly consumption data for a limited number of premises for year 2010
  - Half-hourly load data at feeder level for year 2010

Due to the sensitivity of the information (electricity demand data), the origin of the consumption data was not identifiable; the premise identifiers were removed and only limited information about the premises was available. Such anonymity of the data required finding a way to use it that was still meaningful and as close to reality as possible. This is done as follows: 1) for each substation, a demand profile is selected randomly based on the real billing data and the type of each premise. 2) Each demand profile is then scaled to match the billing data so the average load at each substation is correct, but the shape of the load might change, depending on the selected profiles. Through this approach, the power offered by the simulation of agents at a fine level of detail is somehow lost; however, the general patterns of residential versus commercial consumption are preserved.

Table 3-1 gives an overview of the data types, the number of records and their use in the framework. The data was used differently at the different stages of the simulation. The ABM component uses most of the data described below, either to calibrate the model or to validate it. PSO uses some of the data as is and some after being output by the ABM simulation.

Additional datasets were also outsourced:

- **PV output curves**: example PV generation curves based on typical PV sizes, angles, and efficiency ratings, using TMY (typical meteorological year) data were generated using the PVWatts tool (National Renewable Energy Laboratory, 2012) to be used by the PV agents.
- **Historical Weather Data**: this was purchased from the Bureau of Meteorology, and includes half-hourly rain, temperature, humidity and wind observations (1995-2011), plus 3 hourly cloud observations and daily global solar exposure (1990-2011). This data is used to develop more accurate models of PV output, as well as to generate future consumption profiles based on the correlation that exists between demand and the different weather parameters.

3.5.2 Overview of the modelling approaches

As shown in Table 3-1, most of the data is used by both the ABM and the PSO module; however, their use differs according to the need of the modelling technique.

In the ABM module, each asset that is described in the connectivity data is implemented as an agent. Its characteristics are set, as well as its connection to other agents according to the network hierarchy description. For each agent, a demand agent is also created and referenced so that at each time step of the simulation, the electricity consumed by it (in the case of premises e.g.) or flowing through it (for feeders, e.g.) can be calculated. In the case of generator agents, a supply agent is linked and the electricity produced at each time step is given according to the PV output curves. Following, each simulation run of the model generates ½-hourly demand estimates for every premise and every asset in the distribution network, starting from January 2010 and continuing for a given number of years. Each simulation run can output the following kinds of data, for any collection of network assets:

- A chronological demand curve that covers the entire simulation period.
- An average daily or weekly demand curve for each year (or each season).
- A peak day curve, or peak week curve, for each year (or for each season).

These types of outputs are then transformed to be used by the PSO module. The ½ hourly demand for the peak day for each asset is passed as a matrix to the MATLAB program along with the data associated with distribution and zone substations, distribution feeders, PVs and batteries as well as the required data for converting the line loss and reliability indices to genuine dollar. The PSO outputs a vector of solutions that indicates the mix of network upgrades and installation of DGs that minimize the cost of the network as well as the associated total cost of the network upgrade.

Table 3-1 - Data types used in the framework, and description of their use.

Data Type	Items	Number of items in models	Use in Framework
Network configuration (92,261 asset agents)	Overhead Power Lines (845 km)	5,073	ABM (Calibration) + PSO
	Underground power cables (335km)	1,468	ABM (Calibration) + PSO
	Transformers	2,334	ABM (Calibration) + PSO
	Switches/Isolating Devices	3,945	ABM (Calibration) + PSO
	Premises	79,441	ABM (Calibration) + PSO

Electricity load demand	Billing Data (79,441 premises)	Commercial	5,820	ABM (Calibration) + PSO (transformed)
		Residential	72,471	ABM (Calibration) + PSO (transformed)
		Industrial	1,150	ABM (Calibration) + PSO (transformed)
	½ hourly Metered Data at premises (385 premises)	Commercial	230	ABM (Calibration)
		Residential	87	ABM (Calibration)
		Industrial	67	ABM (Calibration)
	½ hourly Metered data at feeder level		90	ABM (Validation)

3.5.3 Choice of the toolkits for implementation of the framework

The framework presented above is implemented using Eclipse RCP (Rich Client Platform) (McAffer & Lemieux, 2006).

It has MASON (Luke et al., 2005) as its agent-based modelling simulation engine, allowing the platform to be flexible and extensible through the use of Eclipse plugins. A wide range of ABM toolkits were investigated for their suitability to this application, based on the literature (Berryman, 2008; Najlis et al., 2001; Nikolai & Madey, 2009; Railsback et al., 2006) and later on narrowed down according to a set of criteria specific to this project. These requirements specified that the toolkit should: have an open-source license type, use JAVA due to previous experience in software development, be fast in execution and applicable to large number of agents, have qualities of self-organization, adaptation, and causality in networks as defined by (Berryman, 2008), be modular and extensible and be implementable on the Eclipse platform. MASON answered these criteria and was therefore selected and integrated into the Eclipse platform. While MASON provides many functionalities through its GUI, such as the upload of models, visualization of the agents movement and graphs of the outputs, these were decoupled and Eclipse RCP was used for the GUI. Only the simulation engine that contains the methods relating to the scheduling of the agents and those allowing stepping through time were kept for the simulation part.

The particle swarm optimization module is implemented in MATLAB because of its user-friendly programming language designed for numerical computation. It also has several capabilities, such as ease in implementing algorithms and visualizing outputs through the use of its graphing capabilities. It can also interface with many other languages such as C and Java, as well as Excel. This was important to allow the communication with the ABM module as currently this is done using *.csv files generated by the ABM module. Tighter integration of the two modules is a task soon to be implemented. For example, the scenario output by the optimization module will be validated by the ABM, to ensure that the proposed installation of DGs does shave the peak load as predicted.

3.5.4 Demonstration of implemented framework

Having defined the data used in the model, the modelling techniques and the toolkits, the implementation of the platform was possible. As stated above, the framework was implemented using Eclipse RCP and an example is given in Figure 3-2 which shows a part of the distribution network for Townsville, Australia.

The ABM GUI contains many panels which offer different ways of interacting with the model. When opening a model that has been created using the input data described in Section 3.5.1, a tree representation of the network is displayed in the central panel. The different assets composing the network are represented as nodes within a tree, where the edges represent the connection between the elements. Each asset, which node can be selected in the tree view, can have its individual characteristics inspected in the panel on the right side of the screen. Its demand is also displayed as a graph as the simulation progresses, see the right side panels. Any node within the tree can be inspected and its simulation output saved in a file that can be further analysed at the end of the simulation run. This allows finding out those areas in the network that could potentially be at risk of reaching capacity.

The model displayed in Figure 3-2 was created using the data presented in Table 3-1. Therefore 92,261 asset agents were implemented and the corresponding demand either being consumed (for premises) or flowing through (for feeders e.g.)

was associated to these agents. Scenarios were also used with assigning different penetration rates of solar panels which modified the demand for the grid differently over days as well as seasons, using the PV output curves as defined in Section 3.5.1.

Validation of the model was also performed, by comparing the simulated chronological demand curve aggregated at the feeder level with actual measurements for the same feeders which data were provided by Ergon Energy. Figure 3-3 shows the simulated and actual value for one feeder over the month of January 2010. As can be seen, the degree of accuracy varies over the month. The general shape within the days is respected, and some of the days match very closely; also the weekend consumption is lower than the ones during the week. However some peak days for this month have not all been captured which could be due to problems in the definition of the network configuration, where some premises are connected to a feeder other than the one specified in the network connectivity file. Validation of the model is still on-going and as it is being worked on, the model accuracy is gradually improving.

Figure 3-2 - Framework overview – ABM module

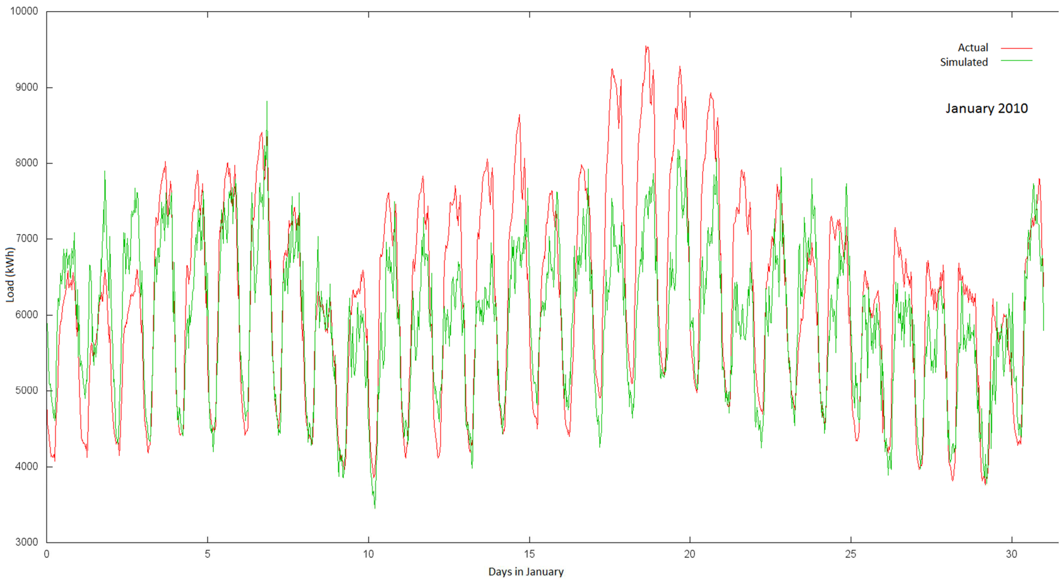


Figure 3-3 - Model validation - Comparison of load data measured at a given feeder and simulation output.

From the simulations, peak consumption days can be identified, which become input to the PSO module in the form of load duration curves. Given this, an objective function is formed which is composed of the network upgrade investment cost (required cost for upgrading substations, feeders, and employed DGs), network reliability cost, and line loss cost. The application of the PSO to the given data provides a vector of particles that gives the optimal placement and sizing of DGs along with the rating of substations and feeders at different years for which the objective function (total cost) is minimized. Several preliminary optimizations have been done for the whole of Queensland, showing that optimal levels of PV can give a cost saving of 1%, battery of 4.1%, while PV plus battery gives a cost saving of 5.5% (Ledwich et al., 2012). Recently, the PSO has been more tightly integrated with ABM so this analysis is now being broken down to finer levels such as individual distribution substations.

In terms of performance, the ABM module can currently run 1.1 million agents per second on Intel i7-2820QM CPU; this means that if the simulation contains 1000 agents, 1100 steps will be performed in a second. It is harder however to estimate the time for the PSO module due to the variation in the parameters for each run. However, for a sample 20-year distribution network planning in presence of DGs and capacitors for a 18-bus test system, it currently takes about 1 hour on a desktop computer with a Core 2 Duo CPU E8600 at 3.33 GHz and 3.46 GB of RAM.

3.6 CONCLUSION

Planning for the most economical and sustainable distribution network with integrating renewable generators is both a challenge and an opportunity for network managers and planners. This multi-level problem (varying temporal and geographical scales) can be answered through the combination of two modelling techniques, agent-based modelling and particle swarm optimization, which takes advantage of each method's strength over given time frames. Their integration in a streamlined manner through the development of a software platform allows both visualizing the dynamic evolution of the system on a fine temporal scale and planning for the least cost placement of renewable generators.

While the two modelling techniques have been successfully integrated into a framework, calibration of the models and their validation through the exploitation of the available data is still on-going. Also, research into modelling techniques for flexible and extensible models using a modular approach is currently undertaken, that will allow extension of the platform capability to other areas of interest as the scope of the project expands. Finally, scenarios will be built to further explore possible futures with the variation of some input parameters reflecting the consequences due to changes for example in policies, population growth in some areas or the impact that demand management measures will have on the premises consumption patterns. Development of such scenarios is key to decision-makers whose better insight into possible futures can help implement more sustainable infrastructure.

3.7 ACKNOWLEDGMENTS

The authors gratefully acknowledge the funding through the NIRAP grant, which is making this research possible, the contributions of diverse partners on this project and especially Ergon Energy for providing the data used in the framework.

3.8 REFERENCES

Australian Government - Clean Energy Regulator. (2012). The Small-scale Renewable Energy Scheme (SRES). Retrieved 02/04/2012, 2012, from http://ret.cleanenergyregulator.gov.au/About-the-Schemes/Small-scale-Renewable-Energy-Scheme--SRES-/about-sres

Australian Government. (2011). Renewable Energy Target. Retrieved 23/08/2011, 2011, from http://www.climatechange.gov.au/government/initiatives/renewable-target.aspx

Berryman, M. (2008). Review of Software Platforms for Agent Based Models. (DSTO-GD-0532). Defence Science Technology Organisation.

Chassin, D. P., Schneider, K., & Gerkensmeyer, C. (2008). GridLAB-D: An open-source power systems modeling and simulation environment. Paper presented at the Transmission and Distribution Conference and Exposition.

Connolly, D., Lund, H., Mathiesen, B. V., & Leahy, M. (2010). A review of computer tools for analysing the integration of renewable energy into various energy systems. Applied Energy, 87(4), 1059-1082. doi: 10.1016/j.apenergy.2009.09.026

del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G. (2008). Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. Ieee Transactions On Evolutionary Computation, 12(2), 171-195. doi: 10.1109/tevc.2007.896686

Ergon Energy. (2010). Network Management Plan Part B: Electricity Supply for Regional Queensland 2010-11 to 2014-15.

Foley, A. M., Ó Gallachóir, B. P., Hur, J., Baldick, R., & McKeogh, E. J. (2010). A strategic review of electricity systems models. Energy, 35(12), 4522-4530. doi: 10.1016/j.energy.2010.03.057

Hall, P. (2011, 24 September 2011). Queensland State Government admits electricity grid failing to cope with solar power systems. Courier Mail. Retrieved from http://www.couriermail.com.au/news/queensland/solar-dream-caught-in-gridlock/story-e6freoof-1226144903889

Kirby, B., & Milligan, M. (2008). Facilitating Wind Development: The Importance of Electric Industry Structure. Golden, Colorado: National Renewable Energy Laboratory.

Komor, P. (2009). Wind and Solar Electricity: Challenges and Opportunities. In PEW Center on Global Climate Change (Ed.). Boulder: University of Colorado.

Ledwich, G., Drogemuller, R., Utting, M., Ziari, I., Boulaire, F., & Abeygunawardana, A. (2012). Planning Future Energy Grids : Renewables - Project Milestone Report (P. Engineering, Trans.) (pp. 1-36). Brisbane, Australia: Queensland University of Technology.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). MASON: A Multi-Agent Simulation Environment. Simulation: Transactions of the society for Modeling and Simulation International, 82(7), 517-527.

Macal, C. M., & North, M. J. (2005, 2005). Agent-Based Modeling And Simulation. Paper presented at the 2005 Winter Simulation Conference.

Macal, C. M., & North, M. J. (2006, December 3-6, 2006). Tutorial On Agent-Based Modeling And Simulation Part 2: How To Model With Agents. Paper presented at the Winter Simulation Conference, Monterey, California, USA.

McAffer, J., & Lemieux, J.-M. (2006). Eclipse Rich Client Platform: designing, coding, and packaging Java applications. Upper Saddle River, NJ: Addison-Wesley.

Najlis, R., Janssen, M. A., & Parker, D. C. (2001, 04-07/10/2001). Software Tools and Communication Issues. Paper presented at the Proceedings of a Special Workshop on Land-Use/Land-Cover Change, Irvine, California.

National Renewable Energy Laboratory. (2012). PVWatts - A Performance Calculator for Grid-Connected PV Systems. Retrieved 05/01/2012, 2012, from http://rredc.nrel.gov/solar/calculators/PVWATTS/version1/version1_index.html

Nikolai, C., & Madey, G. (2009). Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. Journal of Artificial Societies and Social Simulation, 12(2), 2.

Powertech Labs Inc. (2012). DSATools - Dynamic Security Assessment Software. Retrieved 23/05/2012, 2012, from http://www.dsatools.com/

Queensland Government - Office of Clean Energy. (2011). Solar Bonus Scheme. Retrieved 02/04/2012, 2012, from http://ret.cleanenergyregulator.gov.au/About-the-Schemes/Small-scale-Renewable-Energy-Scheme--SRES-/about-sres

Queensland Government. (2009). Securing Queensland’s Energy Future: Regulation for Electricity Demand Management. In Queensland Government - Office of Climate Change (Ed.).

Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. Simulation, 82(9), 609-623.

ROAM Consulting. (2012). 2-4-C Lite. Toowong, Qld, Australia: ROAM Consulting. Retrieved from http://www.roamconsulting.com.au/24CLite/index24C.php

Schutte, S., Scherfke, S., & Sonnenschein, M. (2012, 19-20 April 2012). mosaik - Smart Grid Simulation API. Paper presented at the SmartGreens 2012.

Wikipedia. (2011, 9 August 2011). Agent-based model. Retrieved 23/08/2011, 2011, from http://en.wikipedia.org/wiki/Agent-based_model

Ziari, I., Ledwich, G., Ghosh, A., Cornforth, D., & Wishart, M. (2010). Optimal allocation and sizing of capacitors to minimize the transmission line loss and to improve the voltage profile. Computers and Mathematics with Applications, 60(4), 1003-1013. doi: 10.1016/j.camwa.2010.03.042

AUTHOR BIOGRAPHIES

FANNY BOULAIRE is a Ph.D. student in the Complex Urban System Design group, QUT. She is currently applying agent-based simulation techniques to the modeling of electrical power distribution networks. Prior to this, she worked at CSIRO (2003 - 2011) developing and implementing models for diverse applications related to the urban environment. Her email address is fanny.boulaire@qut.edu.au.

Dr MARK UTTING works for QUT, developing agent-based models of future electricity grids, and is also an Associate Professor in Computer Science at the University of Waikato. From 2009-2011, he also worked for Netvalue.net.nz, using agile techniques to develop Next Generation Genomics Software. He is the author of the book `Practical Model-Based Testing: A Tools Approach', as well as more than 50 publications on model-based testing, formal methods for object-oriented and real-time software and language design for parallelism. His email address is mark.utting@qut.edu.au.

ROBIN DROGEMULLER is the Professor of Digital Design at QUT. He holds qualifications in Architecture and Maths & Computing. He has been working on energy implications of buildings for many years, having contributed to key work leading to the addition of energy performance requirements in the Australian building code. His major research interest is in the integration of computer based analysis tools with architectural and engineering design processes. His email address is robin.drogemuller@qut.edu.au.

GERARD LEDWICH (M'73–SM'92) received the Ph.D. degree in electrical engineering from the University of Newcastle, Australia, in 1976. He has been Chair Professor in Power Engineering at QUT, Brisbane, Australia, since 2000. His interests are in the areas of power systems, power electronics, and controls. Dr. Ledwich is a Fellow of I.E.Aust. His email address is g.ledwich@qut.edu.au.

IMAN ZIARI received his Ph.D. in Electrical Engineering from Queensland University of Technology, Australia, in 2011. He is currently working as a research fellow in Power Engineering at QUT. His interests are in the areas of distribution network planning, renewable energy resources, optimization and power quality. His email address is i.ziari@qut.edu.au.

Chapter 4: Dynamic Agent Composition for Large-scale Agent-based Models

This chapter, written as a journal article[2], presents a novel approach, *dynamic agent composition*, to support the development of flexible and extensible large-scale agent-based models. This approach was implemented in MODAM (MODular Agent-Based Model), a software framework built for planning the future electricity grid using agent-based modelling.

[2] The article was published in the journal of Complex Adaptive System Modelling, 2015.

Dynamic Agent Composition for Large-scale Agent-based Models

Fanny Boulaire, Mark Utting, Robin Drogemuller

Queensland University of Technology, 2 George Street, Brisbane, Queensland 4000, Australia

Dynamic Agent Composition for Large-scale Agent-based Models

ABSTRACT

Purpose

This paper describes *dynamic agent composition*, used to support the development of flexible and extensible large-scale agent-based models (ABMs). This approach was motivated by a need to extend and modify, with ease, an ABM with an underlying networked structure as more information becomes available. Flexibility was also sought after so that simulations are set up with ease, without the need to program.

Methods

The dynamic agent composition approach consists in having agents, whose implementation has been broken into atomic units, come together at runtime to form the complex system representation on which simulations are run. These components capture information at a fine level of detail and provide a vast range of combinations and options for a modeller to create ABMs.

Results

A description of the dynamic agent composition is given in this paper, as well as details about its implementation within MODAM (MODular Agent-based Model), a software framework which is applied to the planning of the electricity distribution network. Illustrations of the implementation of the dynamic agent composition are consequently given for that domain throughout the paper. It is however expected that this approach will be beneficial to other problem domains, especially those with a networked structure, such as water or gas networks.

Conclusions

Dynamic agent composition has many advantages over the way agent-based models are traditionally built for the users, the developers, as well as for agent-based modelling as a scientific approach. Developers can extend the model without the need to access or modify previously written code; they can develop groups of entities independently and add them to those already defined to extend the model. Users can mix-and-match already implemented components to form large-scales ABMs, allowing them to quickly setup simulations and easily compare scenarios without the need to program. The dynamic agent composition provides a natural simulation space over which ABMs of networked structures are represented, facilitating their implementation; and verification and validation of models is facilitated by quickly setting up alternative simulations.

Keywords: agent-based model, dynamic composition, large-scale, electricity distribution network.

4.1 BACKGROUND

Agent-based modelling (ABM) has been used successfully over the last decade to model different aspects of the electricity sector. Its use was initially mainly for the analysis of power market design for large-scale electricity systems when deregulation happened (Batten & Grozev, 2006; North et al., 2002; Weidlich, 2008). These models aimed at investigating the interactions between the physical infrastructure at the transmission level (high voltage networks), and the economic behaviour of market participants to help engineer markets in the electricity sector. The application of ABM to the electricity distribution network (medium and low voltage networks) is not as widespread as that of the transmission networks, but is becoming more studied, especially as new technologies (rooftop solar panels, batteries, …) are appearing on the market and transforming the way electricity is consumed, produced and traded (Cai et al., 2011; Institute for Energy and Transport, 2014).

Agent-based modelling has seen a rise in popularity for its capacity to provide some insight as to how a system responds to changes from the entities' responses and interactions and the environment, by capturing information at a fine level of detail over space and time using simple rules (Klügl & Bazzan, 2012; Macal & North, 2006). It is therefore particularly suited to model the electricity grid which is currently going through a phase of transformation. The way the grid is going to be used is changing, with consumers now also becoming producers and installing new technologies that are changing the flows of electricity on the networks. Communication between the different network assets will become more prominent, impacting further its management but also providing many opportunities. Information about where, how and when these changes are going to happen is important as averages are not sufficient to inform planners appropriately.

Within this context, we have developed a modelling and simulation (M&S) application to answer questions relating to the planning of the future grid and to assess the impact of the integration of decentralised generators (DGs) on a distribution grid owned by Ergon Energy (Ergon Energy, 2013a). This M&S application supports the network planning process by providing an understanding of the evolution of the network in terms of load and voltages, over space and time, and finding the most economical solution in terms of network upgrades. The full platform uses two modelling techniques: agent-based modelling (Castiglione, 2006) and particle swarm optimisation (PSO) (del Valle et al., 2008). The ABM approach was chosen for a few reasons. One is its capacity to capture the information at a fine level of detail both geographically and over time, allowing customers behaviours in relation to their usage of new technologies to be represented and linked to the network structure with accuracy. Another one is that in this application context, the past is no predictor of the future, because very little is known about the impact of the large-scale integration of DGs on distribution networks. By capturing the functioning of the different entities and their relationships to one another, insight into what might happen using simple rules can be gained. In our M&S application, ABM is used to run a large number of scenarios of possible futures, and its output (load duration curves at any node on the network) is passed to the PSO module. This module evaluates which network assets can be installed or upgraded to ensure safe, reliable and economical delivery of electricity. Details on the overall M&S application can be found in (Boulaire, Utting, Drogemuller, Ledwich, et al., 2012).

This paper focuses solely on the ABM part of the M&S application; more specifically on the technical aspect in building a large-scale agent-based model in regards to the requirements of the project. The following requirements were defined, where the model needed to:

a) Take into account the physical characteristics of the network (assets and their connections) as well as the different actors and the environment, influencing the flows on the network;

b) Be able to represent the evolution of the system over many years (long-term planning) but with a fine level of detail that captures how the different elements operate over half-hourly periods;

c) Deal with large and varied datasets coming from corporate databases to populate the model - in terms of configuration of the network and characteristics of its elements, and also allow for different data types for the output of the simulation;

d) Be able to evolve along with the changes in the network and the consumers market

   - With the addition of new technologies over time, as they became available;
   - and different ways in using them, e.g. comparing behavioural or policy impact depending on how the technology is used;

e) Allow creating various scenarios with ease so that simulations can be set up on a daily basis by power engineers, who are not programmers.

Further to these model requirements, goals were identified for its implementation:

- An independent author (a developer that is adding new agents) does not need to modify previously written code;
- The models need to be assembled following a "code-free" approach, where
  - A model user does not need to read or modify the Java code
    - Behaviours of agents can be added/changed without coding – both at the beginning when setting up the model, and during the run,
    - A user can try different models without going into the code, but simply by combining different aspects of the model
  - There is no need for recompiling when adding agents to the model

Two early implementations of our M&S application, based on Repast (Argonne National Laboratory, 2014) and MASON (Luke et al., 2005), exposed various shortfalls with building large-scale models using existing model building approaches in regards to our needs. These are summarised in Table 4-1. The limitations, requirements and goals mentioned above, led to the development of MODAM (MODular Agent-based Model), a framework that builds flexible and extensible large-scale ABMs, using *dynamic agent composition*. This approach consists in having the agents built at runtime by bringing together the physical representation of the elements (assets) and their different behaviours that will specify their actions.

The solutions implemented in MODAM, in response to the shortfalls identified are also given in Table 4-1, and are discussed further in the paper.

Table 4-1 - Shortfalls of existing model building approaches and software systems, and MODAM solutions.

Shortfalls of existing model building approaches and software systems	MODAM Solutions
• Central simulation class is responsible for a. Instantiating all agents; b. Defining relationships between instantiated agents; c. Reading the data used to build agents and relationships (if the model is data-driven); d. Complex option handling done as centralised code.	• Decentralised factories create the physical properties of an agent (assets) and its behavioural properties (behaviours) separately a. Several asset factories create assets and the relationships between them; b. Several behaviour factories attach behaviours to assets; c. Each asset factory can be independently parameterised with data providers; d. Each factory handles its own options.
• Each agent is a single class. a. Variation of behaviour requires many subclasses; b. Each agent is created independently of others.	• Separation of assets and behaviours allows a. Mix-and-match construction of agents at runtime; b. Gathering certain entities in groups.
• Non deterministic order of agents' execution	• Deterministic simulation runs for each random seed (reproducibility of results)

The MODAM framework enforces the separation of assets and behaviours, and combines this key idea with several other techniques (data providers, factories, interfaces, channels (runtime data attributes), use of reflection by a module manager) to fully support the dynamic composition of agents. Because we are building large-scale ABMs where thousands of agents are represented and for which we have sufficiently accurate knowledge, data is used to populate the model, defining the way the agents are in relation to one another as well as their properties. Assets and behaviours are created within components according to their type, and data providers are called on to populate the individuals' entities, with the model coming together at runtime. This facilitates setting up large-scale simulations and has the additional property of not requiring the user to program. This approach builds on the vision set by (Hamill, 2010) of having a library of building blocks for agent-based models. These building blocks would capture a specific environment, or agent, or group of agents and by bringing them together a modeller can set up a simulation more easily, which is especially interesting for the non-programmer. This approach, *dynamic agent composition*, is the key contribution of this paper.

This paper describes this approach, to building flexible and extensible large-scale ABMs. First, the dynamic agent composition is motivated in Section 4.2 using an example of the implementation of an electric vehicle agent. An overview of the approach is then given in Section 4.3, followed in Section 5 by more details describing the asset and behaviour models, and a description of how these elements come together at runtime to create a simulation. Section 4.5 discusses the challenges and the benefits in using this method. Finally, our work is put in relation to other work in the domain of agent-based modelling, and composition.

References to applications of the electricity sector are made throughout this paper to illustrate the use of the *dynamic agent composition* in a concrete manner. More details of the electricity models and simulation results can be found in (Boulaire et al., 2013a; Boulaire, Utting, Drogemuller, Abeygunawardana, et al., 2012; Boulaire, Utting, Drogemuller, Ledwich, et al., 2012).

4.2 MOTIVATING EXAMPLE OF A DYNAMIC AGENT COMPOSITION

Before formalising what is meant by *dynamic agent composition*, this section motivates this approach by introducing a simple example of an electric vehicle (EV) agent. This EV agent is to be used within the context of planning the electricity grid, to understand how the increasing number of vehicles and the way they are used will impact the current infrastructure.

An EV is a mobile agent, which is not bound by its geographic characteristics. It can however have properties of location that will indicate where it connects to the grid to charge. It can be considered as a mobile battery, which can be limited to recharge only, but is also able to discharge to support the premise consumption it is attached to, if needed. It has a state of charge which is the amount of energy that is left in the battery, and from which charging requirements are calculated. Therefore it has similar properties to a battery with a few additional ones.

Following from this, an EV has at least two main properties, from the network viewpoint, that influence its behaviour: driving characteristics, and a charging regime. These need to be accounted for when implementing the rules within the EV agent. Figure 4-1 gives a schematic representation of the composition of an agent, which is made of an electric vehicle asset to which two behaviours are added. The asset includes all the physical and data attributes of the vehicle, such as location, battery capacity, state of charge, maximum charge rate.

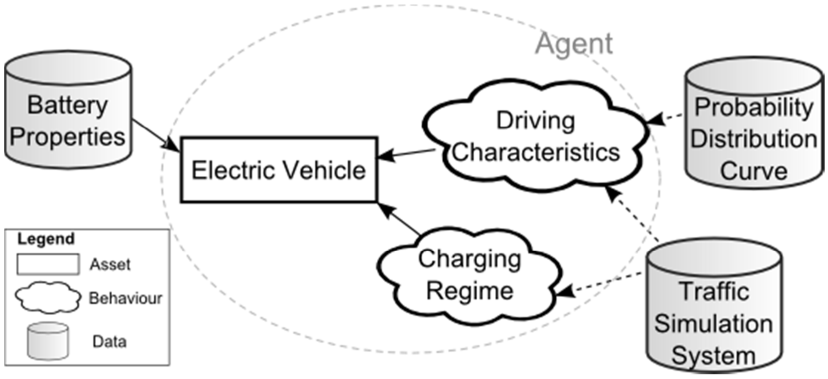


Figure 4-1 - Illustration of the options for an electric vehicle with charging behaviour and driving behaviours.

While these two behaviours need to be described within the rules, each of them can have multiple implementations. For example, we can think of two implementations for the driving characteristics:

- The location parameter is informed by whether or not the vehicle is at a premise, using a Boolean variable that is set to true for the timesteps after which the EV has reached the premise. This time of arrival can be randomly chosen from a probability distribution curve derived from typical home arrival times. Similarly, its charging state is randomly set from probability distribution curves of typical vehicle trips;
- The location parameter is informed by a traffic simulation system that knows where and when the vehicle has arrived. Its charging state is

calculated from the traffic simulation system that knows the exact trip the EV did for the day.

Similarly, we can also have two implementations for the charging regime:

- The EV can charge at any time of the day, as soon as the signal is set for it (e.g. as soon as the vehicle arrives at the premise);
- The EV charges only during set periods of time, which can be defined by a policy setting (set) or informed from communication with a central controller (dynamic).

For each behaviour type, only two options are presented here, but we can imagine many more alternative behaviours.

If the behaviours and the static information were implemented in one same agent class, these alternative behaviours could be implemented by subclassing an electric vehicle agent class for example, or by calling behaviour objects defined in other classes. In this example, this would result in four agents that the user could choose from, which is equivalent to our implementation. However, when increasing the number of behaviours' options, using the dynamic agent composition would result in lesser implementation needs compared to subclassing existing agents implemented in one class.

Having that flexibility in combining the behaviours is important. Comparison of the impact of behaviours of agents is then facilitated by simply swapping a behaviour with another one.

Further, if a behaviour can be used by two asset types, this separation avoids writing extra code and enables reuse. This is the case for the charging regime in our example, which can be used not only to describe charging characteristics of an electric vehicle but also of a battery, whether it is privately owned or is grid operated. This property of reusability is important as it reduces the development time, and the risk of possible implementation mistakes.

In addition to these properties of flexibility and reusability, having the asset and the behaviours separated also offers ease in extending the model. Indeed, there is no need to modify the code of an agent if a new behaviour type is to be added, even if that behaviour is defined by an independent author. This is quite interesting, especially when more information becomes available as the project evolves, or when information/data changes, e.g. due to new applications.

Finally, data is also used to populate the agents' properties, both for assets and behaviours, which offers additional flexibility in the definition of the agents. For example, projections of EV uptake can be used to create x EV assets in year 1 of the simulation, x' in year 2, etc, and additional data to specify the properties of these assets. Behaviours can be associated to these assets following different percentages of expected behavioural profiles of their users passed in the data (e.g. y% of the population are expected to charge at anytime, and y' % at a set time). These behaviour percentages can also be varied to see the impact an incentive might have on the overall network (e.g. y' could be increased to find the percentage at which users should be encouraged to sign up to a tariff incentive, that would benefit the grid).

These different elements are brought together at runtime, creating agents and the agent-based model through a linking mechanism. This dynamic composition of the agents offers great flexibility and extensibility of the ABM, and means that a modeller does not necessarily need to program; they can just combine assets, behaviours and data to create an ABM. Details are given in the following sections.

4.3 OVERVIEW OF THE DYNAMIC AGENT COMPOSITION

This section defines the dynamic agent composition and the different types of compositions that are possible, enabling flexibility, extensibility and reusability in the definition of an agent. Then an overview of how a large-scale ABM is built, using this composition, is given.

4.3.1 Definition of dynamic agent composition

We define *dynamic agent composition* as the process of bringing together at runtime the following distinct entities:

- An asset - the physical properties of an agent (static information);
- One or many behaviours - the rules the asset is subject to, to make its decisions (dynamic information);
- Data

where an agent is defined as:

$$Agent = Asset + Behaviours$$

and the data is used to populate either or both the asset and the behaviour attributes. Data is not a requirement, but offers greater flexibility and facilitates the creation of large-scale models. Combinations of these three entities are then held in a component, or module, for their implementation.

Figure 4-2 shows graphically this dynamic agent composition.

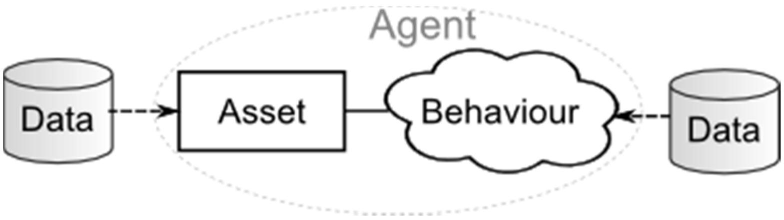


Figure 4-2 - Dynamic Agent Composition

Here, when we mention one asset or one behaviour, we mean one class of asset and behaviour rather than an instance of an asset or behaviour. An asset class will typically have attributes and *setter/getter* methods to populate and access their values, while a behaviour class will contain the rules, held in *start(), step(), stop()* methods.

Data can be used not only to populate the asset and behaviour attributes, but also to determine how many of these are to be created, as well as what their relationship to one another is. Consequently, data reading is not happening at the individual agent level but at a higher level, the factory level, where data is used to populate the individual agents. More detail is given in section 4.4.

This type of breakdown of the agent into asset and behaviour is especially suited to our domain application, where the physical structure of the distribution network, which is quite static, is to be represented along with the way electricity is consumed or flows over it, which is dynamic. While changes in the infrastructure can happen with upgrades and extension of the network, these are quite slow compared to the dynamic behaviour of the usage of electricity which is the model's state variable, whether it is described as a load, a voltage or a current in the simulations. The assets are likely to be used differently for reasons independent of the asset characteristics, although still within their properties' limits, such as when new policies are implemented that affect user behaviours. Being able to easily change the behaviour facilitates their quick assessment on the system as a whole.

Agent Composition Types

Many types of agent compositions can be done, which shows the properties of extensibility, flexibility and reusability in building an agent-based model. Figure 4-3 illustrates many of these compositions.

The base case when extending an agent-based model consists of creating a new agent, which means creating a new asset and its associated behaviours. With the agent composition, an agent-based model can be extended by simply defining new behaviours and applying them to an existing asset, increasing the number of available agent types. This is illustrated with Behaviour B1 for example, which also has alternative implementations (B1', and many up to $B1^n$).

In addition to extensibility, this example illustrates flexibility, as it is possible to choose any of the available behaviours for a new agent type.

Reusability is illustrated with behaviour B2 which can be used by both Asset A1 and Asset A2. In this case, the two assets have very distinct properties; however, one of their distinct features is that they have in common a set of rules to describe an aspect of their behaviour. This could be for example the case for batteries and electric vehicles, which could both be using the same rules for charging control algorithms.

In these three cases of agent composition, an independent author does not need to modify previously written code. New classes can be created, implementing interfaces to specify whether they are assets or behaviours, and these can then be called at simulation setup to compose the required agent.

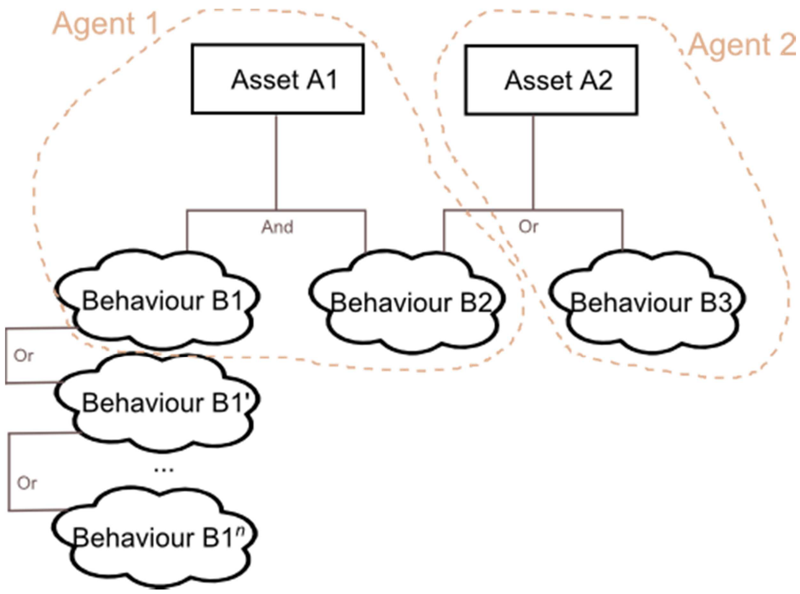


Figure 4-3 - Example of the properties of extensibility, flexibility and reusability when creating agents.

Figure 4-3 shows that an agent is composed of an asset and one or many behaviours. The opposite is also possible, where a behaviour can operate over one or many assets, and update their state at once, within one timestep. Such an example can be found when implementing a global voltage analysis algorithm (load flow (Morton, 2003)) which runs over a group of assets, and updates the assets' voltage at each timestep. This is such that the group of assets over which the analysis is happening needs to be balanced, and therefore have a central place of calculation, considering all the assets at once.

4.3.2 Building the agent-based model

When defining an agent-based model, many agents are created and put in relation to one another to form the system over which they will evolve. Using dynamic agent composition, bringing the different assets, behaviours and data together will define a model. Figure 4-4 illustrates building blocks, or modules, holding these three entities (assets, behaviours and data) where MODAM is the framework that glues them together to form an agent-based model. These modules might contain information relating to many asset or behaviour types at once, or individual ones, depending on the needs. While assets and behaviours are defined individually in their own class, the modules enable them to be grouped together to form sub-systems.

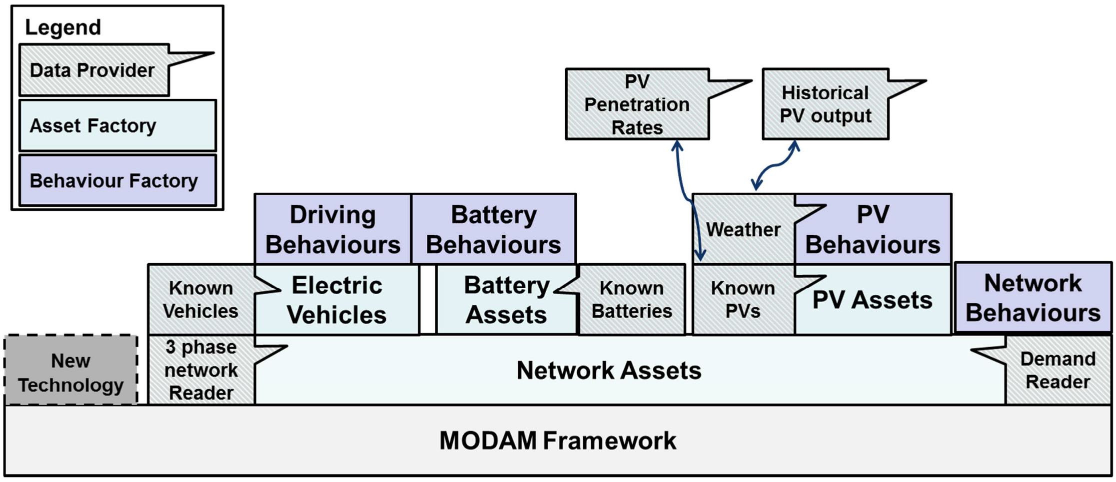


Figure 4-4 - The MODAM framework is the foundation of the ABM model; it connects the different parts of the model. Here a network model describing the network elements and their behaviour is defined. Photovoltaic (PV), electric vehicle (EV) and battery modules have been added to understand how they can support or hinder the functioning of the network. This can be extended in many ways – for example, a 'New Technology' model could be defined and added to the tool to represent any new technology that might impact the network.

In Figure 4-4, we can see four modules that contain information relating to assets describing the network: the network assets module (which contains lines, buses, transformers, switches, etc.), solar photovoltaic (PV), battery and EV assets modules. Four modules are also available for the description of the behaviours, where some can contain many behaviours (e.g. the network behaviours), and others share their behaviours across assets (e.g. the battery behaviour for EVs and batteries). Finally, many datasets are represented that inform the assets or the behaviours, and that can be interchanged or used in combination depending on the need of the model or the availability of the information. For example, for the solar PVs, the assets can be described using known information about their location and their characteristics (that can be used to initialise simulations from the end of the observation period), or using solar PV penetration rates (when making predictions about the future of expected placement and rating of the panels). Similarly, datasets can be used to inform the behaviours, as is the case for solar PV behaviours, which can use a weather model to calculate the output of the solar panels at individual locations taking into account the passage of clouds, or using historical data of solar PV output.

Having described the theoretical framework of the dynamic agent composition, the following section describes how it was implemented.

4.4 IMPLEMENTATION OF THE DYNAMIC AGENT COMPOSITION

Our M&S application was developed using the Eclipse IDE (The Eclipse Foundation, 2012) which is a widely-used open source platform made of a base workspace and an extensible plug-in system for customizing the environment. Using Eclipse on top of OSGi (Open Service Gateway initiative) and Eclipse plugins, which have strong support for modularity, supported our requirement of a flexible and extensible model environment, and was a natural fit to the definition of our components, or modules, which can each be implemented within their own plugin.

This section describes how our agents are created, using different approaches to the implementation of the assets and behaviours models, and how the data is used to populate them. But first, a UML diagram of the main interfaces used in the MODAM framework is presented in Figure 4-5; this diagram will be used to support our explanations throughout this section. As a first introduction, an interface named *IABMState* is at the centre of the framework and holds all the elements for a simulation. It sets a scheduler (*IScheduler*), so has access to the simulation time (*ISimTime*), and has access to assets (*IAsset*) which may have one or many behaviours (*IBehaviour*) associated with them. These assets and behaviours are created by factories (*IAssetFactory* and *IBehaviourFactory*) which are populated by data providers (*IDataProvider*). Each asset has a demand object (*IDemand*) that contains an extensible set of named values that can be set; these are the state variables of the simulation. Finally, the scheduler schedules the behaviours which update the demand at each time step during the simulation. More details for each of these entities are given throughout this section.

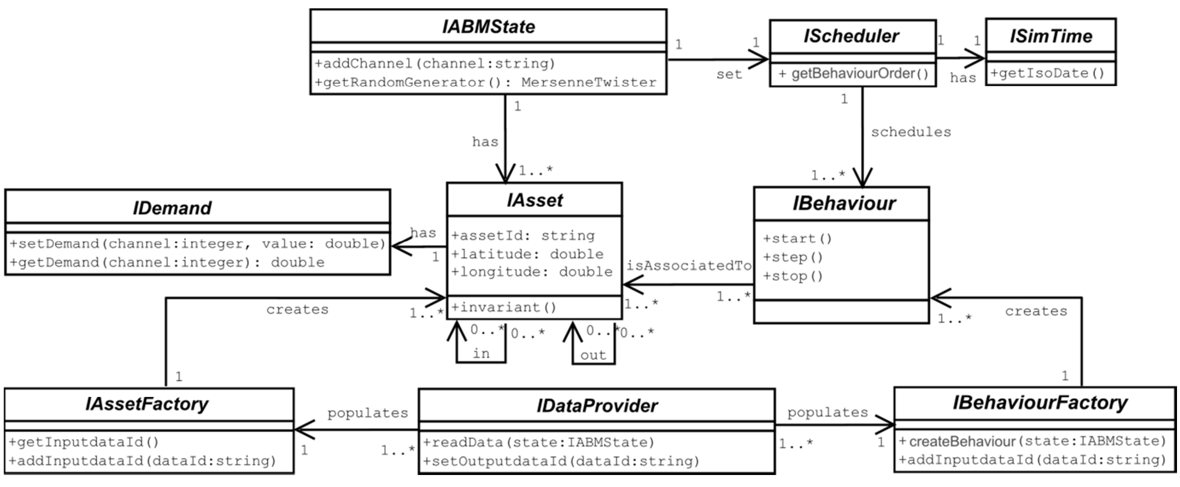


Figure 4-5 – UML diagram of the MODAM framework.

4.4.1 Implementation of an agent

Assets and behaviours are implemented in different ways, according to their requirements.

The Assets Data Model - use of EMF

EMF (Eclipse Modeling Framework (Steinberg, Budinsky, Paternostro, & Merks, 2008)), one of the many available plugins of the Eclipse platform, is a modelling framework that facilitates building code based on structured data models. Particularly well suited to the requirements of our application domain, where the physical infrastructure is to be represented, EMF was chosen to develop a data model describing the assets and their relationship to one another. EMF has the advantage that the implementation of the objects specified in an *ecore* model, described in XMI (XML Metadata Interchange), can have their classes automatically generated, facilitating the implementation of the model within an application.

In addition to this, EMF can handle extension of models, a feature that was of particular interest to us. Any object declared in a data model can be extended or referenced by any other that has been defined in a new child model. Any EMF model can thus be created in a separate plugin and extend one or many models allowing the overall model to keep on expanding. Child models can be in distinct plugins, which makes it possible to choose one model over another at any time, if it describes the problem better, allowing for flexibility in the M&S application.

In our implementation, a few models were defined, which all extend a base model where two entities are defined: *IAsset* and *IDemand*, along with their properties. These entities are implemented as interfaces, as shown in Figure 4-5. Any other EMF model extending this one can then implement these interfaces, and define others as required. One of the main features of the *IAsset* interface is that it contains two properties (in and out) that define an in-out relationship that specify a directed graph over the assets, to represent the networked structure of our domain problem. In addition, each *IAsset* has a string as a unique ID, as well as a longitude and latitude to give its geographical location. Additional attributes can easily be added to assets, or to particular subclasses of assets, simply by extending the EMF model. This generates new subclasses of *IAsset* that have additional private fields plus getter and setter methods. This is highly customisable, but because it involves code generation and compilation, it is best used for relatively static models and is not sufficiently dynamic to support the kind of runtime composition of behaviours that we want.

Where dynamic composition of behaviours is required, a model designer can use the MODAM *data channels* feature, which is provided by an *IDemand* object associated with each asset. This provides an expandable set of named real-valued attributes for each asset. During the model initialisation phase, the behaviour factories register the channel names that they wish to use, and the MODAM framework maps these to integer indices, so the set of variable channels depends upon which behaviours are included in the model. As the model runs, behaviours can read and write the channel values of any asset. This allows behaviours written by different authors to communicate via data channels simply by using a common name for a channel. It also allows MODAM to provide a generic graphing and logging facility that can graph and save any channels from any set of assets. This can be used for visualisation in Google Earth to provide a platform for stakeholders' engagement.

Two EMF models have been implemented so far in our M&S application that both extend this base model, and have been used together to define the overall asset model. The first model contains the different assets that define a base network, as captured in Figure 4-4, in the Network Assets module, as well as solar PV and batteries. The second one contains one asset that describes the way a premise would consume electricity depending on the tariff it is subject to. This second model was implemented to test our hypothesis that it is possible to extend the data model using EMF and that this can be done within other plugins. This implementation was successful and the tariff asset was easily added to the model and integrated within the agent-based model.

The Behaviours – use of the Strategy pattern

Assets and behaviours are implemented independently; however, each behaviour has a reference to its asset in order to retrieve the necessary state variables and make its decisions during a simulation. The implementation of the behaviours' information and rules is contained in the *start()*, *step()* and *stop()* methods of its class that extend an *IBehaviour* interface. The *start()* method belonging to the corresponding asset allows initialising the behaviour, while the *stop()* ends it; the *step()* method updates the behaviour at every timestep.

To obtain flexibility in the behaviour implementation, we used the strategy pattern (Gamma, 2009), using interfaces and defining classes that implement the *start(), step(),*and *stop()* methods. Any number of classes can implement the *IBehaviour* interface, and be called at runtime to specify which behaviour is to be used. One advantage of this approach is that a new plugin can easily define new behaviours, as long as access to the interface is provided; there is no need to access a behaviour class previously defined, only the interface.

4.4.2 Building an agent-based model

Building an agent-based model requires bringing together the assets and the behaviours that form agents and relate them to one another to form the complex system over which they will evolve. This is done within factories, using data from corporate datasets, as explained below.

Factories

Assets and behaviours are created separately, which is done automatically through the use of the *factory* pattern. A plugin can contain one or more asset factories that can control which assets need to be created; many factories can be defined for a given type of asset creation for example, with slight variations depending on the aim of the factory. The same is true for the behaviours.

These factories implement the *IAssetFactory* and *IBehaviourFactory,* depending on whether assets or behaviours need to be implemented. Using the *factory* pattern ensures that the action and interactions of the agents are taken care of in a consistent manner.

To answer our goal of flexibility, each factory typically creates assets or behaviours for one specific type of asset or type of behaviour. This means that each asset defined in an EMF model can implement its own *IAssetFactory*; similarly each behaviour type can implement its own *IBehaviourFactory*. It is however also possible to have a factory that will define many different entities and put them in relation to one another if judged appropriate; this is most likely to happen for the assets only, and is not recommended for the behaviours. An example of this is for the network asset module which contains one factory implementing *IAssetFactory* where lines, buses, switches, transformers and premises are created and related to one another, as assets form the underlying directed graph over which behaviours act.

Factories and data to populate the model

In each of these factories, data can be used to populate the assets and the behaviours. Different types of data can be used, that can define the number of assets to be created, how they are in relation to one another, or what their properties are. One of the requirements of our ABM was that the distribution network be built from corporate data. This means that different types of datasets needed to be handled, and that allowance be made for new types of dataset formats to be input to the simulation. For this, an interface called *IDataProvider* is accessible by both the *IBehaviourFactory* and the *IAssetFactory* during the model instantiation and is implemented by different readers that access various data formats.

Data offers flexibility in setting up simulations as it can be used to represent different areas of study, to compare different trajectories of possible futures by setting different methods of technology uptake, or different methods to describe behaviours. For example a demand behaviour which is associated to a premise and represents its electricity consumption for every half-hour of a day can be set using three types of data in our implementation: half-hourly profile data from a sample of premises, half-hourly profile data from some feeders, and profiles derived from a weather-driven model of consumption. Any of these methods can be chosen to populate the behaviour of a premise consumption individually, and can also be combined using weights to obtain a desired proportion of profile methods.

Depending on the provenance of the data, the format will change, which is handled by different implementations of the same interfaces, providing flexibility in composing the agent-based model. As an example, two types of networks can be used in our current implementation of the agent-based model: a three phase urban network and a SWER (Single Wired Earth Return) network, with data coming from different corporate databases with different formats.

How the ABM comes together

Large-scale agent-based models can be built using dynamic agent composition, either:

- Via explicit Java code, or

- Via command line configuration or GUI, and the use of an automated model builder.

When using explicit Java code, the programmer needs to instantiate the factories, link them to the desired data providers, and execute the factories to build the model. However, this process was automated to answer our goal of code-free construction of agent-based models. This automation is the subject of another paper, but we are giving here an overview of the automation here. A *Module Manager* automatically discovers the plugins and weaves them together. This means that it finds all the available plugins in the registry and enables those chosen by the user. If there are missing plugins, these are found and added automatically. The assets and behaviours are then created with the required data by the Module Manager who instantiates and parameterises the asset and behaviour factories. Each factory handles its own options that have been given in the command line or GUI panel (parameters, and data). Then methods are called on these factories to populate the assets and behaviours with the required data, using reflection, by just knowing the type of interface they implement.

4.5 DISCUSSION

4.5.1 Dynamic agent composition: challenges and responses

While the concept of dynamic agent composition is quite simple, its application to the implementation of large-scale agent-based models with an underlying networked structure has its challenges. These are described below with a discussion about the way we responded to them.

Ordering of the assets' creation

Extending an existing structure, which can be represented by a directed graph, such as the electricity grid where the nodes are assets and the edges their connections, requires a notion of reference. This can be challenging especially as the assets may be defined in separate factories, in separate plugins, and come together to form the complex system only at runtime. For example, if adding battery assets (B), they need to be created and attached to the relevant node (N) (e.g. a premise) in the network. This means that N need to have been created first, and put in relation to the other assets in the initial network. Only then will the battery asset factories be called to create B and attach them to the right nodes N.

To satisfy this requirement of reference over the assets, precedence of the creation of some assets over others was determined. This requires the asset factories to be ordered. If the assets are all created within one factory, their ordering can be handled by the developer within that factory. However, if the assets are defined in independently developed factories, there is a need to mention the order in which the assets need to be created, and consequently the order in which the factories are being called, which can be automatically ordered by the Module Manager. This type of dependency is one of the consequences of aiming at creating an extensible framework, and is the equivalent of inheritance in object programing.

In order to solve this problem, partial ordering of asset factories is used where an attribute (*Predecessors*) allows defining which other factories need to be called before this one. Predecessors that are not included in the current model are ignored, so that maximum flexibility is allowed. For example, if factory F has predecessors A and B, it is possible to run models with any combination of F, A and B. If only A and F are part of the model, then F will automatically be run after A, while B is ignored.

Because of the separation of the *Assets* and the *Behaviours*, this ordering is only necessary on the assets which describe the underlying network structure of the model. Behaviour factories are called after all the asset factories, in any order, since behaviour creations are independent of each other - they communicate only via the assets.

Ordering of the agents' execution

In a "classic" agent-based implementation, agents are often instantiated and scheduled in a central location, e.g. in Repast all the agents are defined in a class implementing *ContextBuilder* (Collier, 2014), and in MASON (Luke et al., 2005) in a class extending *SimState*. This means that it is possible to not only create the agents in a given order but also order the agents execution within a time step in relation to other agents. The same capabilities needed to be available with our dynamic composition. As mentioned previously, ordering the behaviours' creation is not important in MODAM, only the ordering of the assets’ creation is. However, ordering the execution of the behaviours (i.e. within one step of a simulation) is extremely important, and is necessary to enable deterministic simulation results, which is one of our M&S application requirements for verification purposes.

When talking about scheduling of agents, we do not mean that there exists a central planner that will decide on the actions of the agents but rather on the timing of these actions, which corresponds to the execution of the behaviours. The actions themselves are still undertaken in an autonomous manner by the various behaviours. At the start of the simulation, a global scheduler analyses the dependencies between the behaviours and sorts them into a safe execution order.

Each *BehaviourFactory* creates a set of behaviours, and groups them into named *behaviour groups,* which are used to help order the execution of the behaviours within one timestep. One or many behaviour groups can be defined within a single factory. For example, a BatteryBehaviourFactory could create two types of battery behaviours with different battery control algorithms but that can be executed within the same time step with no specific order. In this case, both kinds of battery behaviours will be assigned the same group. However, if one type of behaviours needs to be executed before the other within the same timestep, an order needs to be specified. For example, if the premise batteries need to be executed first, followed by the grid battery to support the network voltage, then two groups, a PremiseBattery group and a GridBattery group, would be created. This would then allow the ordering to be fully managed by the user who can specify when these groups of behaviours need to be executed. The ordering of the behaviour groups is set through an argument in the command line, followed by the name of the behaviour factory.

An example of behaviour group ordering is given in Figure 4-6. It shows three behaviour groups, two of which can be run in parallel (*BehaviourGroup A* and *BehaviourGroup B*), that is with no particular order, with the third one requiring its behaviour to be called after both of them. In each of them we have three behaviours: *BehaviourGroup A* and C have their behaviours ordered sequentially, and *BehaviourGroup B* has two behaviours that are ordered sequentially (b_1 and b_2) and one that can be called anytime (b_3). In the example given in Figure 4-6, the ordering argument in the command line looks like:

-order = (down (BehaviourGroup A)|up(BehaviourGroup B)) ;
up(BehaviourGroup C)

where *up* stands for bottom-up, and *down* for top-down ordering; | shows that *BehaviourGroup A* and *BehaviourGroup B* can be ordered in parallel, and the semi-colon (;) is to show sequential ordering.

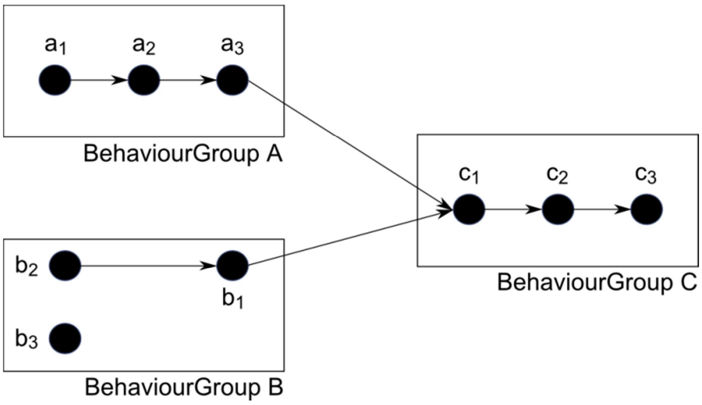


Figure 4-6 - Example of ordering of behaviours within and amongst behaviour groups

State of the agents at a global scale

Because agents can be developed independently and only come together at runtime to create the agent-based model, keeping track of the state variables can be challenging. Indeed, with a fragmented approach, it is expected that different types of state variables are defined by the developers; however, they still need to be accessed within the *ABMState* to allow tracking the state of the system, and also for other agents to be able to access their value to make their own decisions. To remedy this challenge, we used channels, as explained above, which are effectively globally-named and are typically used as state variables of the simulation. This has the additional advantage that being global variables, they do not necessarily need to be state variables but can also be used for other purposes, such as environmental observation (e.g. local temperature, humidity) or globally observable behaviour attributes (e.g. battery charging strategy).

Explosion of the number of assets and behaviours options

The flexibility in separating Assets and Behaviours can have its downfall. It can quickly become very difficult to know what types of assets and behaviours are currently implemented and which ones are able to come together to form meaningful agents. In addition, knowing what types of parameters or data providers are settable from the command line can also quickly become overwhelming.

One of the principles to follow in this case is that, just because it is possible to break down the system into many simpler components does not mean that it should be done. In some cases, it might be advantageous to keep some groups of assets together, or have many rules within one behaviour especially if it will not be reused elsewhere in the future. This concerns mainly the implementation of the factories, and still implies the separation of assets and behaviours, however many assets or behaviours can be created and put in relation within one factory. An example of this, previously mentioned in this paper, is the asset network factory, which creates many different types of assets (lines, buses, switches, transformers, and premises) and puts them in relation to one another. This implementation was chosen because information about these assets and their connections were available at once, and contained in files. Also, within the context of our domain application, this represents the current configuration of the distribution network over which transformations will happen. While a factory was defined in this way, it would still have been possible to create each set of assets independently and create the network using partial order.

Despite using this principle, it is still expected that a large number of factories and datasets will be defined and it can be difficult to keep track of which ones are available. This can be supported by good documentation of the software. To facilitate this, we automated the process of documentation, so that as the model is growing, so is the documentation. Documentation is not only useful for the programmer who would like to add a new plugin for example, but also for the user who will not have access to the code, and who will not want to have to go through it. We have used annotations on asset and behaviour factories and on the methods within the classes that are used for input parameters. These annotations are discovered automatically by the documentation generator, and used to generate consistent documentation for the parameters and required data providers of each factory.

Finally, as the model grows, so will the command line. To prevent having too many parameters to define, and also build on previous simulation runs, it is possible to use a configuration file previously saved, to which additional factories, data providers and parameters are specified. This further extends our goal of flexibility in setting up ABM simulations.

4.5.2 Benefits in using a dynamic agent composition

Benefits in having a clear structure of large-scale agent-based models, through the application of the dynamic agent composition, can be identified from the point of view of the software developers for which it was initially designed. Additional benefits can be identified from the perspective of the clients, or users, as well as for agent-based modelling as a scientific approach. These are discussed below.

Benefits from the point of view of the software developers

The dynamic agent composition was adopted to answer shortfalls initially identified when using existing model building approaches and software systems such as Repast and MASON, see Table 4-1. This approach enabled flexibility and extensibility of both the model definition and its implementation.

Thanks to the distinction between Assets and Behaviours it is easy to change the behaviour of the entities represented in the model. This sometimes needs to happen not only during the model creation but also during the simulation run where they can be added to an existing asset. In ((North & Macal, 2007), chapter 7), the authors explain four model growth paths when building agent-based models: the addition of compatible behaviours, contentious behaviours, compatible agents and contentious agents. These four model growth paths are fairly common, and are supported by the dynamic agent composition approach.

Further, the separation into Assets and Behaviours allows gathering certain entities into groups, when relevant, where all assets that are in relation to one another can be defined at once, while different behaviours can be tried separately over each of them without additional development time. This mix-and-match of entities to create agents can be done at runtime without the need to modify any code; only the command line argument will be altered.

Additionally, having the data separated from the assets and behaviours also makes it easier and faster to extend the model. For example, if a different network is to be modelled, only the asset classes will need to be informed by different data. A new data reader might be implemented if the data format is different; the rest of the code describing the asset and behaviour properties will remain the same. This new data provider will then be called with its associated file at simulation setup, the rest of the command argument remaining the same. This simplifies handling complex options as these are only set in the command line and do not need to be defined in a central class within the code.

Finally, this separation of the agent's aspects into assets, behaviours and data, allows starting an implementation of a model without needing access to all the required information, whether it is data, assets or even a rule that defines the agents' behaviour. This enables an agile implementation (Dingsøyr et al., 2010; Thomas Stober & Hansmann, 2010) of the agent-based model. Also, because the dynamic agent composition is implemented in plugins, it is possible for independent authors to contribute to the model in parallel, which can greatly increase implementation time.

Benefits from the point of view of the users

MODAM was also developed to answer requirements from the users' perspective, which the dynamic agent composition enabled.

Various scenarios can be created with ease by simply changing parameters values, data defining the underlying structure of the network but also behaviours. Having the behaviours independent of the assets makes it easy to add and remove behaviours and assess their impact on the system by setting multiple simulation runs. Further, independent teams can have different implementations of a behaviour, providing customisation of their model to better answer the needs of their analyses. In both cases, there is no need to get into the code; the user can simply bring the building blocks together, by specifying them in the command line.

It is also possible to have a mix of behaviour methods to inform a specific type of assets. For example, if it is expected that a given proportion of the population will behave in a certain way, and the rest in another when using a given asset, these two behaviour types can be applied to the model with varying levels by simply calling on these two behaviours and setting a proportion parameter in the command line. This allocation can be done evenly over the population, or be influenced by known factors such as demographic or geographic characteristics. This has the advantage to represent more accurately how the system might evolve if these proportions are known. When unknown, sensitivity analyses over these allocation levels can be performed to find out what mix would be best for the system. This might be useful for educators, for example, who are trying to bring behavioural change and need to find out the population size to target. Simply varying behaviour calls and parameters values without coding will enable them to quickly set up scenarios.

Finally, the use of data to populate the agents offers the advantage of accurately representing a system, which is not widely done especially for large-scale ABMs, and can be of great benefit for the user. Indeed, many large-scale ABMs are currently developed using taxonomies of agents and probability distributions to represent the system. While such an approach can still be taken using MODAM, we are able to use specific data of the domain of study, which offers the additional benefit of greater fidelity to the system represented. For our electricity network, our asset factories constructed assets based on a data file extracted directly from the Ergon Energy database, so had access to the real attributes of each asset, and their connection to one another. This was a requirement of our client, who is interested in knowing as accurately as possible what is likely to happen on their network at specific locations.

Benefits from the point of view of ABM as a scientific approach

Taking the dynamic agent composition approach when developing MODAM also highlighted advantages in terms of agent-based modelling as a scientific approach.

The separation of an agent into asset and behaviours creates a natural simulation space over which ABMs of networked structures are represented. Indeed, the assets and their connections form a complex network representing the overall structure, which becomes the space over which the behaviours interact with one another. In many simulations such as the Schellings' model (Schelling, 1971), a grid is defined as a 2D matrix, over which the behaviours will evolve and get information to make their decision. Here, the idea is similar where the environment is represented by a scale-free network made of the assets which are publicly available within the model and allow every entity to know their relationship to one another. The behaviours are not bound by structure directly but access the underlying network through their asset. The behaviours only contain private data on which they make their decisions. The assets hold the state variables of the ABM simulation which are modified by the behaviours as they make their decisions. While the assets' connections form a network of their own, references to their geo-location (longitude and latitude) are maintained, allowing displaying the network using spatial information software.

MODAM allows replicability of simulations results, thanks to its deterministic capability through each random seed. While independence of the execution of the agents is still ensured through randomisation of their decisions output, having a deterministic order of their execution allows replicability of the experiment and reproducibility of the results.

Finally, MODAM facilitates model comparisons and validation of behavioural sub models. Indeed, data can be used to set the parameters of sub models, that some behaviours use to inform their decisions. As an example, our implementation of solar PV behaviours can be informed by historical solar PV output data or weather data, see Figure 4-4. The weather data is used to populate a weather-driven model of expected electrical output of solar PV subject to weather with the passage of clouds, while the historical solar PV data simply gives the electrical output of specific solar PVs recorded over a period of time. Because these two approaches require different data input, two data readers were written to be used by each method. When the user chooses their preferred behaviour method, the required data provider will then be called upon. This has the advantage of extending the model, but also offers model comparison and validation of the behavioural sub models. Indeed, the weather-driven solar PV output model could be validated by comparing its output with the actual observations of solar PV output. The dynamic agent composition therefore has the additional benefit that validation of models can be done easily by setting two simulations and simply changing the datasets and data providers, and comparing their outputs.

Finally, MODAM was developed in Java using open-source frameworks (Eclipse, EMF) and standards (OSGi), ensuring that it is soundly constructed, but also enabling it to be reproduced or extended by interested users.

4.6 RELATED WORK

Agent-based modelling, a bottom-up modelling technique, describes autonomous agents and their relationships at a fine level of detail with the view of capturing the dynamics of a complex system (Bonabeau, 2002; Macal & North, 2010). Many toolkits are available to support the development of agent-based models (Berryman, 2008; Luke et al., 2005; Najlis et al., 2001; Nikolai & Madey, 2009; North, 2013; Railsback et al., 2006) with most containing the following features: agents, a scheduler, an interaction space, random number streams, logging and a user interface (North, 2013). In these toolkits, agents are generally made up of a unique identifier, behaviours that can be activated and attributes that can be modified. Both the static information and the behaviours are then held in one place, often defined within a class, because object-oriented programing is well suited for agent-based modelling implementation. While held in one class, however, behaviour implementations might still be the result of the composition of behaviours extending others, as is the case for example for the JADE architecture (Bellifemine, Caire, & Greenwood, 2007), where these are extending behaviour classes (Bellifemine, Caire, Trucco, & Rimassa, 2010). However, these behaviours still are to be added within an agent's implementation, whose architecture is partly hidden, and which requires coding to define the agent. The dynamic agent composition presented in this paper distinguishes itself from these ones as behaviours can be combined without the need to access the agent's code. The behaviours are combined with an asset to form an agent, which can be done by a non-programmer, through a command line, and this composition of the agent happens at runtime.

Our dynamic approach of composing the agents rather follows some of the principles described in MALEVA (Briot & Meurisse, 2006) where composability of behaviours is described. While similarities exist between our conceptual frameworks, with behaviours being composed to form a more complex one, they differ in some aspects. MALEVA uses connectors and has output interfaces so that the output of one behaviour is the input of another one to form a chain of complex behaviours. This is not our chosen approach as we are rather more interested in alternative behaviours. While we can use many behaviours to compose one, we do not have this element of precedence of the way the behaviour is executed. Further, our composition concentrates on bringing asset characteristics and behaviours together to form an agent rather than bringing behaviours together, whose need arose from the requirements of growing models, especially when dealing with large-scale ABMs. Because our behaviours communicate via the assets, we also have more flexibility to mix-and-match of the different combinations than MALEVA. Our approach is rather closer to the one specified in (Bae, Lee, & Moon, 2012) where a hierarchy of models composed of an action model, an agent model and a multi-agents model has been defined. This is done so that agent-based models can be built incrementally and in a flexible manner, which goal is the same as ours. The authors have presented a formal specification using DEVS (Discrete Event System Specification) formalism (Zeigler, 1976), which they have applied to show that two models could be formalised independently and brought together. It is unclear however that more than two models could be brought together easily, and how large the agent-based models can be. Also, to our knowledge, there is currently no implementation platform to support this specification.

Growing large-scale ABMs is not well documented in the literature, which mainly concentrates on the speed of execution of simulations rather than the modelling needs. However, Parry, in (Parry, 2012) differentiates two problems when increasing the scale of a multi-agent system, which includes the computational resources but also the increase in difficulty with a growing agent-based model. Despite this distinction, most of the paper however focusses on how to deal with large-scale simulations which suggests optimising existing code, considering simple solutions such as upgrade of the hardware or evaluating the suitability of the chosen scaling solution on a simplified version of the model. Other approaches to dealing with large-scale simulation requirements is the use of alternative computational techniques, such as considering distributed or parallel implementations of the agent-based model. To this end, simulation toolkits now offer parallel and distributed implementations of their initial implementations as is the case for Repast and MASON for example (Collier, 2013; Cordasco et al., 2013), amongst many others. These allow running larger simulations while still getting reasonable execution time, as described in (Parker, 2007), where an epidemic simulation runs up to 6 billion agents using a distributed simulation. While the focus of this paper is not on large-scale simulations but rather large-scale model, such challenge is also at the heart of our problem. For this, we have implemented a parallel implementation of our ABM scheduler, as described in (Boulaire et al., 2013b), however, this is not the subject of this paper.

4.7 CONCLUSION

This paper has defined *dynamic agent composition*, a novel approach to build large-scale ABMs. This approach had for goal to extend, with ease, an agent-based model with an underlying networked structure. Also, it aimed at having it flexible so that many scenarios could be created using large corporate databases, without the need for a programmer to build the simulations. By breaking down the model into components containing the data, the assets and the behaviour descriptions, and providing a mechanism to bring them together at runtime to compose the agent-based model, this was achieved.

This approach has many advantages over the way agent-based models are traditionally built for the user as well as the developer. Developers can extend the model without the need to access or modify previously written code; they can develop groups of assets and behaviours independently and add them to those already defined to extend the model. The model can then evolve as new agents need to be modelled, which facilitates the models to be used and extended after previously-defined goals are modified. Users can mix-and-match already implemented asset and behaviour components to form large-scales ABMs. This allows them to quickly setup simulations and easily compare various scenarios without the need to program.

Further, using data extracted from corporate databases to populate the ABM enables to represent accurately the physical infrastructure over which the agents evolve. This aspect of our approach contributes to the application of ABMs to the electricity domain as most ABMs use taxonomies or probability distribution to represent the network under study.

Future work includes continued expansion of the model to include more asset types as well as behaviours of assets' usage, such as small-scale generators other than solar PVs, and electric vehicles with different charging methods. While currently applied to the electricity distribution grid only, it is expected this approach can be used more broadly and be of benefit to other applications, especially those that have a networked structure, such as water or gas networks.

4.8 COMPETING INTERESTS

The authors declare that they have no competing interests.

4.9 AUTHORS' CONTRIBUTIONS

FB: has been lead author in drafting the manuscript, has designed and implemented the dynamic agent composition method, has conducted experiments, and has revised the paper. MU: has made key contributions to the paper conception, has aided in the design of the dynamic agent composition method and co-implemented it. RD: has reviewed and revised the paper for important intellectual content. All authors read and approved the final manuscript.

4.10 ACKNOWLEDGEMENT

The authors gratefully acknowledge the funding through the NIRAP grant, which is making this research possible, the contributions of diverse partners on this project, and especially Ergon Energy for supplying us with data of their network and guidance in terms of tools requirements.

4.11 REFERENCES

Argonne National Laboratory. (2014). The Repast Suite. Retrieved 11/11/2014, 2014, from http://repast.sourceforge.net/

Bae, J. W., Lee, G., & Moon, I.-C. (2012, 2012). *Formal specification supporting incremental and flexible agent-based modeling.* Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.

Batten, D. F., & Grozev, G. (2006). NEMSIM: Finding Ways to Reduce Greenhouse Gas Emissions Using Multi-Agent Electricity Modelling *Complex science for a complex world: exploring human ecosystems with agents* (pp. 227-252). Canberra: ANU E Press.

Bellifemine, F., Caire, G., Trucco, T., & Rimassa, G. (2010). JADE Programmer's Guide. Boston, MA , USA.

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Hoboken, New Jersey, USA: Wiley-Blackwell.

Berryman, M. (2008). *Review of Software Platforms for Agent Based Models*. (DSTO-GD-0532). Edinburgh, South Australia, Australia: Defence Science Technology Organisation.

Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America, 99 Suppl 3*(3), 7280-7287. doi: 10.1073/pnas.082080899

Boulaire, F., Utting, M., & Drogemuller, R. (2013a, 18-26 May 2013). *MODAM: A MODular Agent-based Modelling Framework.* Paper presented at the 2nd International Workshop on Software Engineering Challenges for the Smart Grid as part of the 35th International Conference on Software Engineering (ICSE 2013), San Fransisco, CA, USA.

Boulaire, F., Utting, M., & Drogemuller, R. (2013b, 26/08/2013). *Parallel ABM for electricity distribution grids: a case study.* Paper presented at the 1st Workshop on Parallel and Distributed Agent-Based Simulations, Euro-Par 2013, Aachen, Germany.

Boulaire, F., Utting, M., Drogemuller, R., Abeygunawardana, A., Ledwich, G., & Bell, J. (2012, 6-7 December 2012). *Planning for the Impact of Distributed Solar Energy on the Grid.* Paper presented at the 50th Annual Conference, Australian Solar Energy Society (Australian Solar Council), Swinburne University of Technology, Melbourne.

Boulaire, F., Utting, M., Drogemuller, R., Ledwich, G., & Ziari, I. (2012, 9-12 December 2012). *A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network.* Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.

Briot, J.-P., & Meurisse, T. (2006). A Component-based Model of Agent Behaviors for Multi-Agent-Based Simulations. In P. Stone & G. Weiss (Eds.), *Proceedings of the 7th International Workshop on Multi-Agent-Based Simulation (MABS'06), 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'2006)* (pp. 183–190). New York, NY, USA: Association for Computing Machinery (ACM).

Cai, C., Jahangiri, P., Thomas, A. G., Zhao, H., Aliprantis, D. C., & Tesfatsion, L. (2011, 24-29/07/2011). *Agent-Based Simulation of Distribution Systems with High Penetration of Photovoltaic Generation*, San Diego, CA.

Castiglione, F. (2006). Agent based modeling. *Scholarpedia*, *1*(10), 1562. doi: doi::10.4249/scholarpedia.1562

Collier, N. (2013). Repast HPC Manual. Retrieved 01/09/2014, 2014, from http://repast.sourceforge.net/docs/RepastHPCManual.pdf

Collier, N. (2014). Interface ContextBuilder<T>. Retrieved 03/01/2015, 2015, from http://repast.sourceforge.net/docs/api/repast_simphony/index.html

Cordasco, G., Chiara, R. D., Raia, F., Scarano, V., Spagnuolo, C., & Vicidomini, L. (2013). *Designing computational steering facilities for distributed agent based simulations*. Paper presented at the Proceedings of the 2013 ACM SIGSIM conference on Principles of advanced discrete simulation, Montreal, Quebec, Canada.

del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G. (2008). Particle Swarm Optimization: Basic Concepts, Variants and

Applications in Power Systems. *Ieee Transactions On Evolutionary Computation, 12*(2), 171-195. doi: 10.1109/tevc.2007.896686

Dingsøyr, T., Dybå, T., & Moe, N. (2010). Agile Software Development: An Introduction and Overview. In T. Dingsøyr, T. Dybå & N. B. Moe (Eds.), *Agile Software Development* (pp. 1-13). Berlin, Germany: Springer Berlin Heidelberg.

Ergon Energy. (2013). Corporate profile. Retrieved 02/06/2013, 2013, from https://www.ergon.com.au/about-us/who-we-are/our-company/corporate-profile

Gamma, E. (2009). *Design patterns: elements of reusable object-oriented software*. Boston: Addison-Wesley.

Hamill, L. (2010). Agent-based modelling: The next 15 years. *Journal of Artificial Societies and Social Simulation, 13*(4), 7.

Institute for Energy and Transport. (2014, 18/07/2014). Agent Based Modelling for Smart Grids. Retrieved 20/07/2014, 2014, from http://ses.jrc.ec.europa.eu/agent-based-modelling-smart-grids

Klügl, F., & Bazzan, A. L. C. (2012). Agent-based modeling and simulation. *AI Magazine, 33*(3), 29-40.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). MASON: A Multi-Agent Simulation Environment. *Simulation: Transactions of the society for Modeling and Simulation International, 82*(7), 517-527.

Macal, C. M., & North, M. J. (2006, December 3-6, 2006). *Tutorial On Agent-Based Modeling And Simulation Part 2: How To Model With Agents.* Paper presented at the Winter Simulation Conference, Monterey, California, USA.

Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*(4), 151-162.

Morton, A. (2003, 27-30/09). *A fast 'do-it-yourself' load flow algorithm for power systems with sparse topology*. Paper presented at the AUPEC 2003 Australasian Universities Power Engineering Conference, Christchurch, New Zealand.

Najlis, R., Janssen, M. A., & Parker, D. C. (2001, 04-07/10/2001). *Software Tools and Communication Issues.* Paper presented at the Proceedings of a Special Workshop on Land-Use/Land-Cover Change, Irvine, California.

Nikolai, C., & Madey, G. (2009). Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation, 12*(2), 2.

North, M., Conzelmann, G., Koritarov, V., Macal, C., Thimmapuram, P., & Veselka, T. (2002). *E-laboratories : agent-based modeling of electricity markets*. Paper presented at the American Power Conference, Chicago, IL (US).

North, M. J. (2013). A theoretical formalism for analyzing agent-based models. *Complex Adaptive Systems Modeling, 2*(1), 3-3. doi: 10.1186/2194-3206-2-3

North, M. J., & Macal, C. M. (2007). *Managing Business Complexity*. New York, NY: Oxford University Press.

Parker, J. (2007, 2007). *A flexible, large-scale, distributed agent based epidemic model*. Paper presented at the 2007 Winter Simulation Conference, Washington, DC, USA.

Parry, H. R. (2012). Agent Based Modeling, Large Scale Simulations (pp. 76-87). New York, NY: Springer New York.

Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. *Simulation, 82*(9), 609-623.

Schelling, T. C. (1971). Dynamic Models of Segregation *Journal of Mathematical Sociology, 1*, 143-186.

Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008). *EMF: Eclipse Modeling Framework* Boston, MA, USA: Addison-Wesley Professional.

The Eclipse Foundation. (2012). About the Eclipse Foundation. Retrieved 27/02/2012, 2012, from http://www.eclipse.org/org/

Thomas Stober, & Hansmann, U. (2010). Overview of Agile Software Development *Agile Software Development: Best Practices for Large Software Development Projects* (pp. 35-39). Berlin, Heidelberg: Springer Berlin Heidelberg.

Weidlich, A. (2008). *Engineering interrelated electricity markets: an agent-based computational approach*. Heidelberg: Springer [distributor].

Zeigler, B. P. (1976). *Theory of modelling and simulation*. New York, N.Y: Wiley.

Chapter 5: Large-scale Agent-based Modelling and Simulation - Automation Based on a Dynamic Agent Composition

In this chapter[3], a process of automation to build large-scale agent-based models is explained. This process is based on the dynamic agent composition described in the previous chapter, where agents, whose aspects are defined in distinct components, come together at runtime to create an agent-based model. The Module Manager, which weaves the components together in an automated manner, is described using formal specifications written in Z.

Thanks to this automation, large-scale ABMs can easily be composed by a non-programmer who can quickly set up simulations of various scenarios.

---

[3] This chapter's content was used as the basis for a conference paper presented at the 2nd international workshop about sets and tools (SETS 2015).

Specification and Validation of the MODAM Module Manager

Mark Utting[1] and Fanny Boulaire[2]

[1]University of the Sunshine Coast, Australia and The University of Waikato, New Zealand

[2]Queensland University of Technology, 2 George Street, Brisbane, Queensland 4000, Australia

Large-scale Agent-based Modelling and Simulation - Automation Based on a Dynamic Agent Composition

ABSTRACT

Electricity distribution networks are large complex systems that are continuously evolving, both in terms of configuration, technology and management, which modelling is well suited to methods from the domain of complex adaptive system such as agent-based modelling. This paper introduces MODAM (MODular Agent-based Model), a software framework developed to support building large-scale agent-based models (ABMs) for electricity distribution network planning where the models are assembled in an automated manner at runtime.

This automation of the model building for large-scale ABMs is the subject of this paper. It is based on the principle of modularity and composition where the different agents, described in modules, come together at runtime in an automated manner through composition. This paper describes this process, and discusses the pros and cons of such a method. While specifically developed and tailored to study problems in the electricity sector where agents evolve over a networked structure, MODAM is applicable to other domains.

Keywords: Agent-based modelling, automation, modularity, composition, networked structures.

5.1 INTRODUCTION

Electricity distribution networks are undergoing rapid changes with the introduction of new technologies, policies and demand management options. As an example, rooftop PV installations have increased dramatically over the last few years in Australia from an initial capacity of 23 MW in 2008 to around 3,017MW at the end of October 2013 (AEMO, 2012; Australian PV Institute & Australian Renewable Energy Agency, 2013), changing the load patterns on some parts of the low and medium voltage networks, especially in areas of high uptake. This trend is expected to continue, bringing with it other technologies such as battery systems to support further its use and potentially completely transforming the electricity sector. For example, a study done by the Future Grid Forum, predicts that batteries in combination with energy efficiency, gas generation and solar panel could lead a third of the customers to leave the grid in Australia by 2050 (CSIRO Future Grid Forum, 2013). Understanding how changes in one part of the network can affect the rest of the network is important to better plan the future grid. For this, planning tools used by decision-makers need to take into account the new technologies and new approaches that constitute the many possible options that will impact the future grid.

We have developed a simulation environment, called MODAM (MODular Agent-based Model), to assess the impact of different trajectories of consumption at varying locations of the network over many years. It supports understanding the changes in load at every node in the network with the introduction of new technologies (e.g. solar panels, batteries), new policies (e.g. time-of-use tariffs) or/and demand management. It was developed using agent-based modelling (Castiglione, 2006) which was chosen for its capacity to represent at a fine level of detail the behaviours and interactions of the network's entities that have a spatial and temporal component, as well as the behaviours of the different consumers impacting it. Because in this application the past is no predictor of the future, using this bottom-up technique was further beneficial for its representation of the actions and interactions of these entities using simple rules (Klügl & Bazzan, 2012). Agent-based modelling has been successfully used for different applications in the electricity sector (Batten & Grozev, 2006; Cai et al., 2011; Institute for Energy and Transport, 2014; North et al., 2002; Weidlich, 2008), for properties such as those mentioned above, and is particularly suited to our application domain.

Ergon Energy (Ergon Energy, 2013a), a Queensland electricity distribution company, commissioned this project to perform simulations of the future of their grid. To answer their needs, the following software requirements were drawn:

1. A large-scale agent-based model is to represent the Ergon's distribution network, with data coming from corporate databases. This is to represent the current state of the network with accuracy, where thousands of entities stored in very large datasets are to be used to populate a base model;
2. From there, trajectories of consumption are to be simulated using different assumptions and knowledge about the type of technology that will likely be taken up, their location and the way it is going to be used;
3. The model can grow and change easily over the time of the project and beyond it, bringing extensibility and flexibility in the model definition;
4. A vast range of scenarios needs to be tried easily, and this can be done on a daily basis by an engineer, without the need to code.

To support building flexible and extensible large-scale agent-based models, we have developed a dynamic agent composition. It consists in breaking down an agent as an asset and a set of behaviours, where these aspects are defined separately in the model definition and its implementation. These two aspects then come together at runtime only, to form what is traditionally called an agent (made of attributes and behaviours (North, 2013)). This method facilitates building an agent-based model incrementally, as well as offers many options for agents to be defined using alternative or combining behaviours.

With this dynamic composition approach, a manual approach to bringing the agents together at runtime into a coherent agent-based model can be taken. However, this requires a modeller to code, which does not answer our last requirement of code-free set up of simulations. Consequently, this process was automated, which is the subject of this paper.

A Module Manager weaves together modules that hold information about the agents, to compose a model at runtime, in an automated manner. This approach therefore is based on the principle of modularity and composition. Three aspects to support this automated process can then be distinguished:

- A technical aspect to support modularity - it defines the information held in modules, and the technology that will support building the software in a modular manner to support the composition of a model at runtime;
- A collaborative aspect between the user and the MODAM framework to support composition - it specifies which modules as well as what data are to come together at runtime;
- An automation aspect - where a module manager weaves the different assets, behaviours and data to create an agent-based model, on which simulations are run.

These three aspects are the focus of this paper, especially the third one.

The remainder of this paper is organized as follows. Section 5.2 presents the types of simulations that can be done using MODAM, and motivates the automated approach we have chosen to build agent-based models, illustrated by an example. Section 5.3 then presents the three aspects to support the automated process mentioned above, where the technology used to implement our system is introduced in section 5.3.1. Section 5.3.2 introduces how large-scale ABMs can be composed by introducing the command line, and the GUIs. Then section 5.3.3 explains the automation of the composition by formally specifying the Module Manager, using Z (Hayes et al., 1986). Section 5.4 discusses the challenges and benefits in assembling an ABM in an automated manner, and explores an example where our approach was applied to another domain. Finally, section 5.5 reviews the literature, looking at automation of the composition of agent-based models, and more broadly composition in software engineering and simulation environments.

5.2 OVERVIEW OF MODAM

This section describes the types of simulations that can be performed using MODAM. An example of a model is also given, that describes a distribution network to which rooftop solar panels and batteries have been added. While a small network is shown for illustration purposes, it is scalable to large-scale models where thousands of agents are represented.

5.2.1 Types of simulations performed using MODAM

The simulations performed with MODAM are discrete time simulations, where the model is built in terms of logic and the simulation time is represented as a discrete variable. It is using synchronous (time-stepped) time advance mechanisms. Additionally, the agents are heterogeneous agents, as they are required to perform different roles and are subject to different objective functions. An example of agents are consumption agents that respond to the electricity needs of a premise's inhabitants – these vary depending on the function of the premise (residential, commercial, industrial) as well as the environmental conditions (weather conditions). Another example of agents are the battery agents that can have different goals depending on whether their function is to support the grid voltage if grid-operated, or to minimise reliance on the grid if privately-owned.

There are many types of state variables produced by the simulation that inform the state of the network: load, real and reactive voltages, and real and reactive currents at each node in the network. These are calculated for every half hour over a given period (as chosen by the user, from one day to 20 years). The number of agents varies from one simulation to another; however, the framework is set as a large-scale ABM where thousands of agents can co-exist. Simulations with up to 200,000 agents have been run using this platform.

Population dynamics emerge from the combined behaviour of the individuals. The individuals' behaviours are represented by rules for describing for example the voltage drop because of the resistance of a line, or the power injected to the grid depending on the available energy of a battery and the rating of the inverter. Adaptation and fitness-seeking are not modelled explicitly, but they are included in the rules. Individuals know their own characteristics which are influencing their own output values, as well as the environment they are in and the other agents they are interacting with. Interactions are modelled explicitly at the individual level – each entity knows to which one it is connected to and which environment it is in. Stochasticity in the simulation is obtained through the allocation of load profiles for individuals, as well as location of some new technologies such as PVs and batteries. Load profiles from actual records were used as input to the model and allocated randomly to the consumers located at the leaf of the network. The random allocation was however guided according to some criteria so that it was as close to reality as possible – for example recorded loads were allocated to the type of premises it came from, e.g. residential premises as opposed to commercial ones.

5.2.2 Agent-based model of a small distribution network

Figure 5-1 shows an extract of a network that can be modelled in MODAM, on which simulations can be run. The different assets making the distribution network are represented along with the premises, where solar panels and batteries can also be attached to. While this graph shows an integrated network, it was built in three distinct phases with the network assets and premises modelled first, on which simulations were first run. This model then had the PVs added, and finally the batteries in a third phase.

These phases derived naturally from the process of developing an agent-based model, where elements to be modelled are identified over time and the model grows incrementally. In our case, our project started with the need to represent the current state of the network, to which small-scale renewables are integrated to the grid. As time went on, this model further needed to add batteries as their use became more prominent. It is expected that other technologies that might not yet be commercialised, will also need to be added later to the model. Consequently, a very strong focus on our ABMS environment development was the need for extensibility to allow building the model in an incremental manner, with a particular emphasis on allowing agents to be added easily and placed within the network as groups while still maintaining their individual characteristics of placement and properties. This grouping is illustrated in Figure 5-1, where all the agents of the same type (e.g. solar panels, batteries) are added at once to the model, while still having distinct properties (e.g. residential vs. industrial premises, and specific load patterns, etc…).

While these phases illustrate a growing model, they can also be used to highlight the impact of a particular technology by setting up alternative scenarios. For example, three scenarios can be compared here where the first one is a baseline scenario, the second one has additional PVs, and the third one batteries and PV. These can highlight the impact of one technology over the others, taking into account the interactions of the different agents. Simply adding or subtracting the effect of a technology might not reflect the actual inter-dependencies of their usages, which are captured through the relations of the agents.

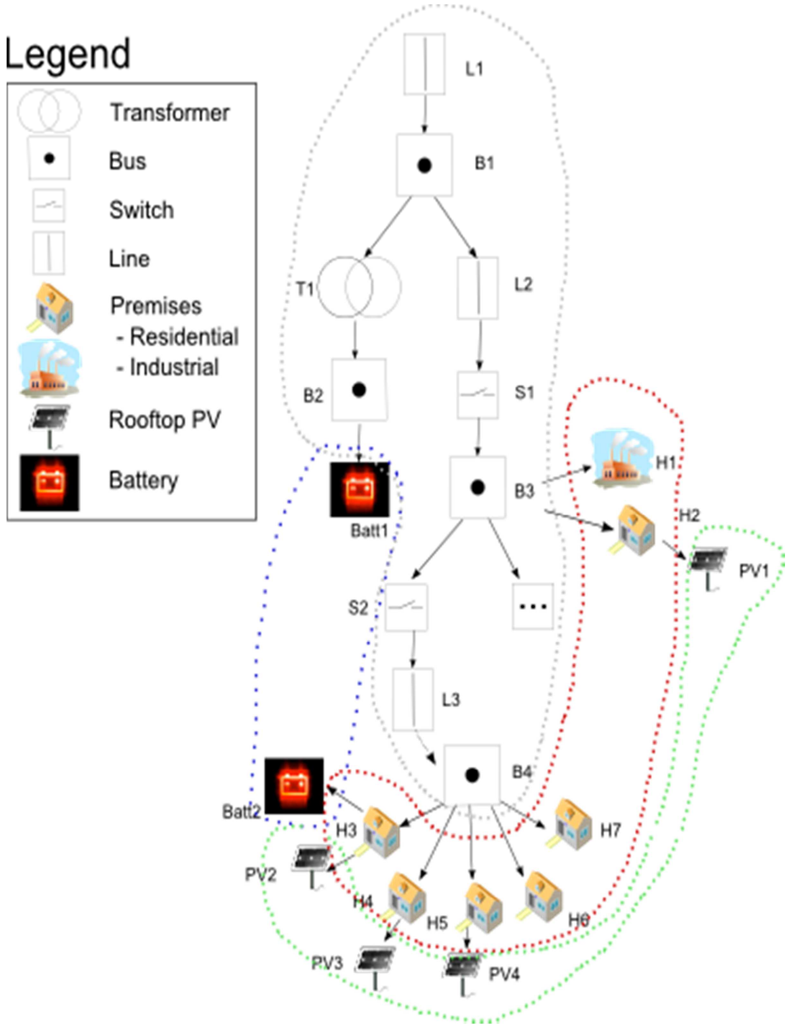


Figure 5-1 - Small extract of a distribution network, containing network assets, as well as solar panels and batteries.

Further to this extensibility property, flexibility in the choice of behaviours to describe the agents was necessary. Indeed, depending on the way the new technology is being used and especially when applied to large numbers of users, very different patterns can be observed on the flow of electricity on the grid. Understanding how some settings can impact the grid is important, especially when assessing the effectiveness of some policies. Further, depending on who the user of a technology is, its usage might vary greatly.

This is the case for batteries whose algorithms describing their usage will vary depending on the user's goal. For example, one battery control algorithm could discharge and recharge over set periods of time every day, where the battery's discharging load covers the needs of the asset it is attached to. Such battery usage would be representative of one of a family wanting to reduce their dependence on the grid when the price of electricity is higher, in the evening, for example. Another battery control algorithm could learn from the load patterns of the previous day to identify times of higher load requirements that need to be clipped. Such algorithm would be useful for a business who wishes its load to remain under a threshold to avoid penalties, as negotiated in their retailer's contract. Finally, another algorithm could get the load information higher up the network (e.g. at a transformer), so that the battery discharges as much as possible over a period of time in order to clip the load aggregated at that level. This algorithm would be used by a network distribution company who owns a battery or has access to a pool of privately owned smaller ones, to clip the load they see at a transformer so that it does not get overloaded for example, avoiding it to overheat and lose life. Changing the battery control algorithm in different simulations can highlight the impact of their introduction and management options.

In our example, in Figure 5-1, two types of batteries are installed, a grid operated one (Batt1 under Bus B2) and a privately-owned one (Batt2 under Premise H3). While both agents are created within a same group, their behaviours differ. To answer both these requirements of extensibility and flexibility we developed an approach which we have called 'dynamic agent composition'.

5.2.3 Overview of the dynamic agent composition

The dynamic agent composition is defined as the process of bringing together at runtime information about an asset, one or many behaviours and data such that an agent is defined as:

$$Agent = Asset + Behaviours$$

And data can be used to populate either or both the asset and behaviours characteristics.

This definition has been successfully applied to our model. Figure 4-4 gives a schematic representation of the different building blocks, also called components or modules that are currently available in MODAM. For our example in Figure 5-1, we would need to use the modules of Network Assets, PV assets and Battery Assets to which Network Behaviours, PV and Battery Behaviours would be added which could vary depending on the type of asset being represented in the network. Data could also be used to inform the characteristics of the assets or the behaviours, and can be held in different formats and come from different sources.

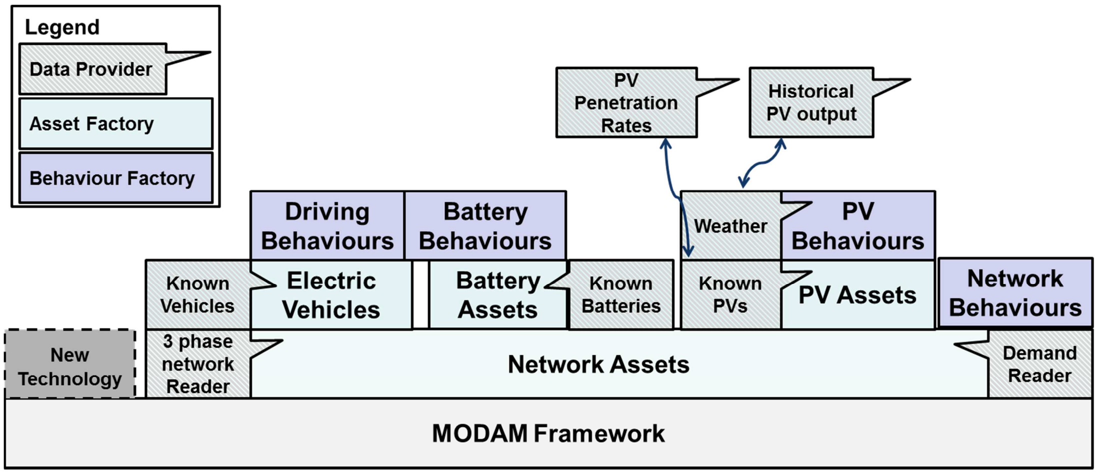


Figure 5-2 - The MODAM framework is the foundation of the ABM model; it connects the different parts of the model.

Based on this dynamic composition approach, factories were used to define both the assets and the behaviours separately, populated by data. To implement an agent-based model using this approach, a programmer can instantiate the factories, link them to the desired data and execute the factories. However, this implies coding, which does not answer our last requirement of code-free set up of simulations. Consequently, this process was automated, as explained in the Section 5.3.

5.2.4 Simulations over the small network

Once the model built for a specific scenario, a simulation can start. It is then possible to observe the state variables over time over the network. Depending on the display of the output, data can be saved as *csv* files to be used as input for further analyses, e.g. for statistical analysis, data mining; or the output data can be saved in SQL databases and further converted to *kml* files for certain features to be displayed in Google Earth.

By visualising output of simulations in Google Earth, a planner can quickly see which part of their network is likely to be overloaded over time. Then, the planner can drill down to those most-at-risk assets, and further investigate them to understand how a particular asset, for example a transformer or feeder, is impacted.

An example of the output of a simulation exported to Google Earth is given in Figure 5-3. In this picture, one same transformer is being observed over time as the simulation runs, showing how it gets overloaded. This asset's load can then be investigated further, looking at its peak load graph to understand how it is modified according to the usage of other types of assets influencing electricity flows. This helps the planner to make decisions about whether or not to upgrade the transformer.

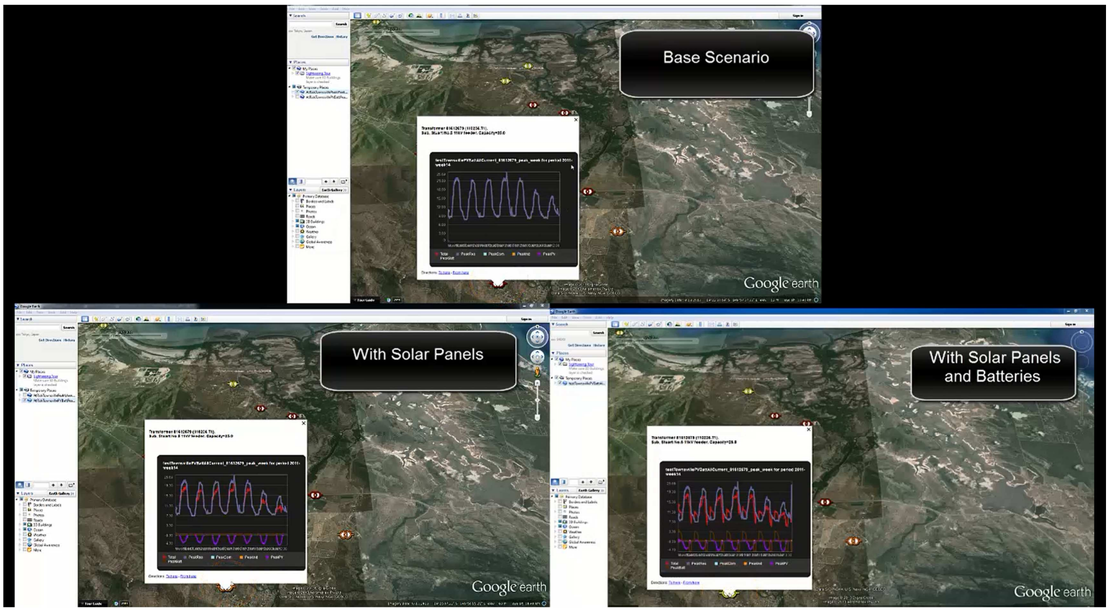


Figure 5-3 - Snapshot comparing three simulations output.

When drilling down to a specific asset, the analyst can then compare the impact of the technologies using other means such as graphs extracted using data stored in *csv* files.

Figure 5-4 shows such graph of the loads for the peak week of the final year of the simulation. It shows more clearly that the baseline scenario indicates that the transformer will be overloaded for a few hours 4 days during the peak week. With the introduction of 10% of solar panels (scenario 2), only one day sees the load over the rating of the transformer (25kW); also running the simulation over time shows that PV helps delay the upgrade of the transformer for a year, as the supply of electricity manages to reduce the peak until the final year. However, this is only temporarily as the peak carries on after the sun has set; and only with the introduction of batteries discharging from 5pm to 8pm can the overload be avoided (scenario 3).

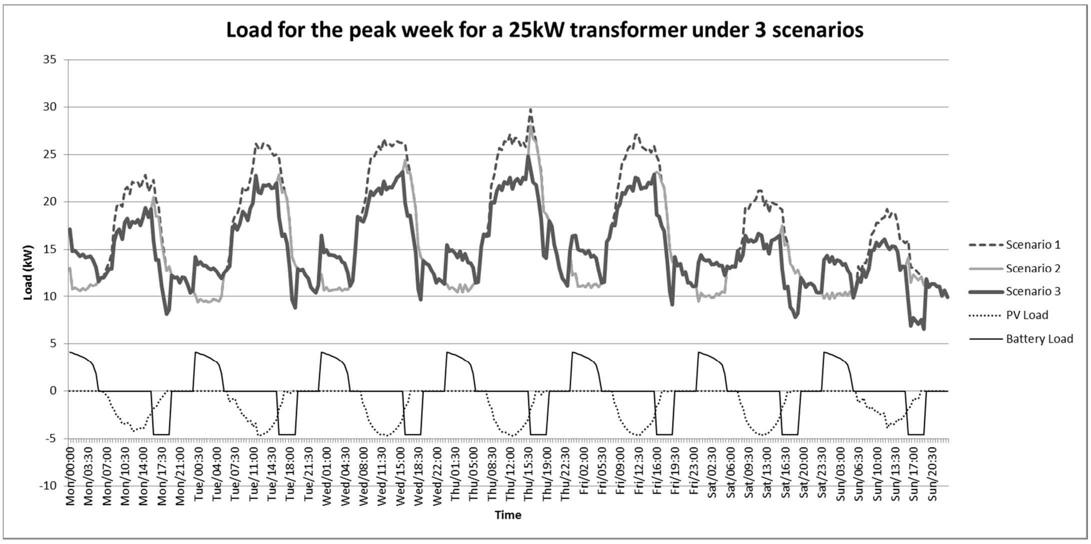


Figure 5-4 - Comparison of three scenarios for the peak week of a 25kW transformer.

5.2.5 Comparison of MODAM implementation features to common features of ABM libraries

To summarise, MODAM is a simulation environment for agent-based models that enables assessing the impact of different trajectories of consumption at varying locations of a distribution network over many years. The way it is built follows the same principles found in most ABM libraries, as shown in Table 5-1 where a comparison of the MODAM approach in respect to North's Modular Imperative Architecture (MIA) Features (North, 2013) is given.

MIA Features	MODAM Implementation features
Agents (ID + attributes +behaviours)	Attributes can be represented by Java object fields or by named channels within each agent. Behaviours are represented as separate Java objects that can be dynamically associated with agents.
Scheduler	The scheduler is time-stepped.
Interaction Spaces	Latitude and Longitude are attributes of the agents to define their physical space; A directed network is available to define connections between the agents.
Random Number Streams	Uniform Mersenne Twister is provided.
Logging	Data channels for a given set of agents can be logged into *csv* files or into databases.

User Interfaces	MODAM provides command line support and two GUIs: a generic GUI and an electricity-network-specific GUI.

Table 5-1 - Comparison of MODAM to North's Modular Imperative Architecture (MIA) Features (North, 2013).

One distinguishing features that sets MODAM apart from most libraries is the way the models are built, using *dynamic agent composition*, where agents which have been defined as an asset and a set of behaviours, come together to form an agent at runtime only. These two aspects of the agent are defined within distinct classes and factories are used to create groups of assets and behaviours of the same type, which can be informed by data. Only when the user declares the factories of assets, behaviours and the data providers that are required to form the agents and put them in relation to one another will the agent-based model be built. There is then no need to program to put the agents in relation to one another, as this can be done in an automated manner, which is the subject of the next section.

5.3 AUTOMATED PROCESS TO BUILD LARGE-SCALE ABMS

This section describes the three aspects that are necessary to support the automated process of building large-scale ABMs, based on the dynamic agent composition: the technology used to support modularity, the way the components are specified to form a model using composition, and finally how these components come together in an automated manner at runtime using a Module Manager.

5.3.1 Technology used to support modularity and specifications of a module

Use of Eclipse and OSGi

One of the main criteria when building our ABMS application was for it to be a flexible and extensible model environment. This meant that the choice of technology on which to implement the model was critical, as it needed to support a modular approach. For this, we used Eclipse on top of OSGi and Eclipse plugins which have strong support for modularity.

OSGi (formerly Open Services Gateway Initiative) (OSGi Alliance, 2013) is a specification that enables writing modular software where the information is encapsulated into software components. The modules in OSGi are called bundles, which have the properties of being installed, started, stopped or un-installed at run-time. This means that applications using OSGi result in being very modular and dynamic, where a bundle can be modified without the need for the system to be rebooted. This specification has been used by the Eclipse Community, whose plugins are OSGI bundles (Vogel, 2012) and are the smallest units of modularisation. OSGi was thus chosen as the technology to support modularity and the code was implemented in plugins within Eclipse.

A module, which is implemented as a plugin, can be defined as:

$$Module = Name + Assets + Behaviours + Data$$

which follows the definition of the dynamic agent composition. From this definition, three types of plugin content can be distinguished in the current implementation of MODAM: some that hold the data information, others the assets information and again others the agents' logic (i.e. the behaviours).

Using Eclipse plugins architecture, a schema can be created where extension points are defined so that any other plugin that will extend the initial plugin will contribute one or many extensions. For our application, three extension points are defined in the main plugin of the MODAM framework: *AssetFactory*, *BehaviourFactory* and *DataProvider*, that all have a class name as a parameter. In addition, the *DataProvider* has a unique identifier and a path parameter to indicate the default file to be used; both *AssetFactory* and *BehaviourFactory* have a set of identifiers that can be linked to a reader to populate their objects.

As many plugins as necessary can be created within MODAM. Within each plugin, extensions of any of the three extension points can be declared. A plugin can contain any of these three extension points simultaneously but it is recommended to keep separate plugins for each of them to increase the flexibility of the model. However, within a plugin, the same extension point can have many implementations of its classes that can be chosen by the user depending on the need of the simulation.

This specification of extension points and extensions is specific to Eclipse. Each new plugin will have its *plugin.xml* file that can be edited by the programmer who will define the data providers and factories to be implemented and how they relate to one another.

Formal specification of a Module

Eclipse gives access programmatically to the information defined in the *plugin.xml* file through the platform registry. Access to the classes and parameters values defined in the file then enables executing the desired actions by putting these entities in relation adequately. In our case, this process consists in bringing the different components together so that the assets and behaviours properties can be accessed to form an agent-based model.

To facilitate this, we created a Module class which we populated using the information contained in the plugin definition. A Module has the following types [*ClassName, ExtId, DataId, Path, Contributor, JavaType*] where:

- *ClassName* is the set of all possible Java class names,
- *ExtId* is the set all possible Eclipse extension point identifiers (for example, au.edu.qut.modam.assetfactory),
- *DataId* is the set of all possible Data identifiers - these are used to link providers and consumers together via named ports,
- *Path* is the set of all file system paths to input data files,
- *Contributor* is the identifier of a module, and
- *JavaType* is a set of all possible Java parameter.

Further, to these types we have

- *MethodName == DataId* where MethodName can have any prefix to DataId and set as the same type;
- *dataProviderError : ClassName* which is an error for the data providers*;*
- *BOOLEAN ::= YES* | *NO* which is a boolean type, with *YES* == 1 and *NO*== 0

A Module has the following definition in MODAM:

$$
\begin{array}{l}
\textbf{Module} \\
\hline
contributor : Contributor \\
extensionClass : ExtId \leftrightarrow ClassName \\
enabledClasses : \mathbb{P}\, ClassName \\
consumes : ClassName \twoheadrightarrow\!\!\!\!\!\!+ \; \mathbb{F}\, DataId \\
produces : ClassName \twoheadrightarrow\!\!\!\!\!\!+ \; DataId \\
path : ClassName \twoheadrightarrow\!\!\!\!\!\!+ \; Path \\
enabled : BOOLEAN \\
prior : ClassName \twoheadrightarrow\!\!\!\!\!\!+ \; \mathbb{F}\, ClassName \\
assetFactories, behaviourFactories, dataProviders : \mathbb{F}\, ClassName \\
\hline
dataProviders = \mathrm{dom}\ produces \\
assetFactories = \mathrm{dom}\ prior \\
behaviourFactories = (\mathrm{dom}\ consumes) \setminus assetFactories \\
\mathrm{dom}\ consumes \subseteq \mathrm{ran}\ extensionClass \\
dataProviders \subseteq \mathrm{ran}\ extensionClass \\
enabledClasses \subseteq \mathrm{ran}\ extensionClass \\
\mathrm{dom}\ path \subseteq dataProviders \\
enabled = NO \Rightarrow enabledClasses = \varnothing \\
dataProviders \cap assetFactories = \varnothing \\
dataProviders \cap behaviourFactories = \varnothing \\
assetFactories \cap behaviourFactories = \varnothing \\
\hline
\end{array}
$$

The *contributor* field is used to access the OSGi bundle but the name will be sufficient.

The *extensionClass* maps each extension point identifier to all the class names that extend that extension point.

The *enabledClasses* is the set of classes that have been enabled within this module.

The *consumes* field maps every factory (*AssetFactory* and *BehaviourFactory*) to its allowable input data identifiers. Note that the implementation goes the other way around, which assumes that each input data id is required by only one class within each module. The domain of *consumes* is made of *BehaviourFactories* and *AssetFactories* that are distinct classes, and that make the whole set of *ClassNames*.

The *produces* maps every data provider class name to the data id that it produces.

The *path* maps each data provider class name to its default path (if one has been specified).

The domain of *produces* is the set of all data providers. The domain of *consumes* is the set of all factories. The domain of *prior* is the asset factories, and the remaining ones in *consumes* are *BehaviourFactories*.

The *AssetFactories*, *BehaviourFactories* and *DataProviders* are distinct sets of class names.

The union of the range of *prior* is within the domain of *prior*, which is the set of *AssetFactories*.

enabledClasses are within the range of the extension classes, that have been enabled by setting the boolean value.

5.3.2 Composing flexible ABMs

Creating a model is done in a collaborative manner between the user and the MODAM framework. There are currently two main ways of interacting with the ABM simulation software: command line scripts or graphical user interfaces.

Command line scripts

The command-line scripts are typically used to run a sweep of scenarios on a cluster of computers. This allows many alternative scenarios to be explored at once. Or it can be used to run the same scenario many times, with different random seeds so that the people and premises in the simulation have different behaviour each run - then the results of all those runs can be analysed statistically to determine the averages, standard deviations, and Probability of Exceedance (PoE) 50 and PoE 10 levels of demand peaks for each part of the network. There is no need to know about programming, simply knowing which factories have been implemented and are available is sufficient. For some of the selected factories, additional arguments and datasets can also be specified that will be used to populate the ABM and set some simulation parameters. An example of a command script is given in Code 1.

Code 1 – Example of a command line for a simulation containing a base network, with solar panels and batteries.

```
+M=assetnetwork                                                              Network
        +C=assetnetwork.ergon.NetworkAssetFactory
        +C=assetnetwork.agent.TariffAgentFactory
        +C=assetnetwork.agent.DemandSumFactory
+M=assetreader
        +C=assetreader.NetworkReader
        +C=assetreader.LocationReader
+M=demandreader
        +C=demandreader.historical.HistoricalDemandReader                    Demand
        -DGrowthRate=1.05
        +C=demandreader.billing.BillingDataReader
+M=pvasset
        +C=pvasset.PVAssetFactory                                            Solar
        -DAllocMethod=R                                                      Panels
        -DValue=10
+M=pvagent
        +C=pvagent.WeatherPVAgentFactory
+M=pvasset.reader
        +C=pvasset.reader.assetcharacteristics.PVAssetCommonReader
+M=weatherreader
        +C=weatherreader.CloudDataReader                                     Weather
        +C=weatherreader.TemperatureDataReader
+M=battery.asset
        +C=battery.asset.BatteryAssetFactory                                 Batteries
+M=battery.asset.reader
        +C=battery.asset.reader.allocation.BatteryExactAllocationReader
        -DSourceLocation=../Resources/exactBattery.csv
        +C=battery.asset.reader.assetcharacteristics.BatteryAssetCommonReader
        -DSourceLocation=../Resources/CommonBatteriesGrid.csv
+M=battery.agent
        -DDischargeTimes=17-20
-GraphSaver=jdbc:googleearthproject:postgres:password
-Graph[TOP/Transformer]=peak-week                                            Output
-from=2010-01-01            -to=2015-01-01
-seed=1234                                                                   Simulation
-output=../Resources/testTownsvillePVBattAllCurrent                          Parameters
-order=up(TariffAgent+WeatherPVAgent+DemandSum);up(BatteryTimeAgent);up(DemandSum)  Agents
                                                                             Ordering
```

The command-line script above describes the information required for a simulation of a base electrical network, on top of which solar panels and batteries have been added, such as the one presented in Figure 5-1. The boxes surrounding the commands highlight groups of information, within which modules, classes and parameters are defined.

The modules are specified with the command "+M" followed by the name of the plugin, and their required classes "+C" followed by the names of the classes (*AssetFactory*, *BehaviourFactory* and *DataProvider*) that are to be used within those plugins. Each of the specified classes can also be parameterised using the “-D” command and their parameter value. The Module Manager will find the modules in the registry and then instantiate the classes specified; reflection is used on the parameter name to set its value within the class's method. Other parameters for the simulation run are then passed. These are the start and end times of the simulation, called using “-from” and “-to”, the “-seed” for the random number generator to ensure reproducibility of simulation experiments, and “-output” for the folder that will contain the output of the simulation. At last, a parameter “-order” is also given which is used by the scheduler to schedule the agents at each timestep in the given order.

While many modules may be available, not all of them need to be loaded, only those required for a given analysis type. However, as the model grows, many modules might be required which can lead to a very long command line. To prevent having too many parameters to define, and also build on previous simulation runs, it is possible to use a configuration file previously saved, to which additional modules, classes and parameters are specified.

Graphical User Interfaces (GUIs)

We have developed two types of GUIs: a generic one and one tailored to the needs of distribution planners, which is a web-based user interface.

The generic GUI provides all the available plugins, factories and datasets to the user. This means that as the number of plugins increases, this GUI also extends without the need to code it further. The new plugins are automatically detected from the Eclipse registry and added to the list of available modules, along with the required parameters definitions. Despite the advantages this GUI offers in terms of flexibility and extensibility when new plugins are created, some of its disadvantages are its lack of domain knowledge, which means that meaningless combinations could be chosen, as well as its flat module structure, that does not offer grouping of related options.

The tailored GUI, on the other hand, only uses a limited number of plugins and factories, selected to answer predefined questions. It is not as flexible as the generic one because a programmer will need to add widgets to call the newly created plugins as they become available. However it is more user-friendly for a planner who does not need to know what factories to select and only wants to perform a limited number of analysis types, where the data populating the models and the simulations parameters can still be varied. An example of the client part of the interface is shown in Figure 5-5 - this typically runs in a web browser and communicates with the main ABM program that is running on a server computer. The left-hand screenshot shows how a segment of the network can be selected and viewed geographically. Then the Demand, PV and Battery tabs are used to set up a simulation scenario, and the Run tab (right-hand screenshot) can be used to run the simulation (on the server) for a given time period. The scenario is also saved so that it can be used as a basis for command-line batch simulations later. The input files and output files of scenarios are all stored on the server, so they can be shared between several people, but maintained centrally. In addition to being user-friendly, this GUI has the advantage to enforce consistency in the selection of the plugins, factories and datasets, preventing meaningless combinations. Each tab knows when its options are complete and indicates it to the user by changing the aspect of the tab, for rapid identification of the simulation selection.

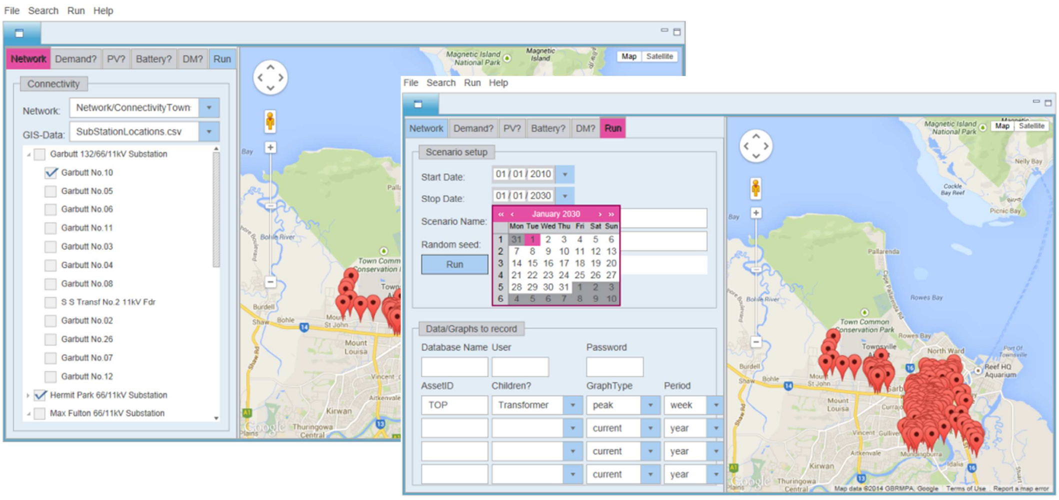


Figure 5-5 - Web-based user interface for running agent-based network simulations.

5.3.3 Assembling a model: an automated process within MODAM - Formal specifications of the Module Manager

The Module manager, as its name indicates, manages the modules specified by the user in the command scripts or the GUI. To assemble the model and start the simulation, the Module Manager follows the below steps:

1. Find all the modules in the registry, collect the enabled modules and extension points;

2. Check and warns of missing dependencies ;
3. Instantiate one instance of each factory and data provider and process their *SetterGetter* methods (see definition of *SetterGetter* below);
4. Call the asset factories, in the correct order using topological sort, to create the assets and populate them with data if specified;
5. Call the behaviour factories to create the behaviours and populate them with data if specified;
6. Call the start method on all the behaviours;
7. Start stepping the simulation.

The following formally specifies the Module Manager class and its different methods that enable the agent-based model to be built in an automated manner at runtime.

1.1.1.1 Definition of the Module Manager class and its methods to initialise, add modules and get missing dependencies

The Module Manager has the following definition:

$$
\begin{array}{l}
\underline{\quad ModuleManager \quad} \\
modules : Contributor \mathrel{⤀} Module \\
outputid : ClassName \rightarrow DataId \\
\hline
(\forall\, n : dom\ modules \bullet (modules\ n).contributor = n) \\
outputid = \bigcup \{ n : dom\ modules \bullet (modules\ n).produces \} \\
\hline
\end{array}
$$

The *ModuleManager* has a set of modules which map a *Contributor* to a *Module*, and of *outputid* that map a *classname* to a *DataId*. Each contributor is uniquely associated to a module. An *outputid* is the union of all the contributors which have a set of data providers.

$$
\begin{array}{l}
\underline{\quad MMInit \quad} \\
ModuleManager \\
\hline
modules = \{\} \\
\hline
\end{array}
$$

At initialisation, a set of empty modules is created. Modules are then added, as the contributors are being identified from the registry.

$$
\begin{array}{|l}
\textit{AddModule} \\
\hline
\Delta ModuleManager \\
m? : Module \\
\hline
modules' = modules \cup \{m?.contributor \mapsto m?\} \\
\hline
\end{array}
$$

Once the modules specified by the user have been discovered and enabled, their *AssetFactories*, *BehaviourFactories* and *DataProviders* can be linked to one another, so that the assets and behaviours can be created and populated by the required data.

1.1.1.2 Factories and Data providers creation

Factories are used to create the individual assets and behaviours of any type. These assets and behaviours can be informed by data to set the number of entities to create, or some of their parameters or in the case of the assets to set their relationships to one another. Depending on the information and the format available in the files, different *DataProviders* are available, and these are linked to their factories by the Module Manager. For this, a SetterGetter method has been defined; it is used by the factories to populate the assets and behaviours with data.

$$
\begin{array}{|l}
\textit{SetterGetter} \\
\hline
name : MethodName \\
argType : JavaType \\
value : ClassName \\
optional : BOOLEAN \\
\hline
\end{array}
$$

It has a name, an argument type (one only), a value which modifies the class and an optional field.

A factory is then defined as follows:

$$
\begin{array}{|l}
\textit{Factory} \\
\hline
name : ClassName \\
methods : \mathbb{P}\, SetterGetter \\
\hline
\end{array}
$$

A *Factory* has a class name and a set of *SetterGetter* methods, so that the relevant data can be used to populate the model.

In order to assign the data to their factories, the dependencies between the factories and the data providers, which are defined through their *DataId*, are checked. *MissingDependencies* are defined as follows:

$$
\begin{array}{|l}
\hline
\textit{MissingDependencies} \\
\Xi \textit{ModuleManager} \\
\textit{missing}! : \mathbb{P}\, \textit{DataId} \\
\hline
\textit{true} \\
\hline
\end{array}
$$

A *getMissingDependencies* operation that finds all the non-optional data requirements that are not satisfied by any of the enabled data providers is then called by the Module Manager. If any missing dependency is returned, the current model setup is incomplete, meaning that the model cannot be run.

The *getMissingDependencies* operation is defined as:

$$
\begin{array}{|l}
\hline
\textit{getMissingDependencies} \\
\Xi \textit{ModuleManager} \\
\textit{factories}? : \mathbb{P}\, \textit{Factory} \\
\textit{missing}! : \mathbb{P}\, \textit{DataId} \\
\hline
\textit{missing}! = \{ \textit{unsatisfied} : \textit{DataId} \mid \\
\quad (\exists\, m : \mathrm{ran}\ \textit{modules};\ f : \textit{factories}?;\ s : \textit{SetterGetter} \bullet \\
\quad\quad m.\textit{enabled} = \textit{YES} \land \\
\quad\quad f.\textit{name} \in m.\textit{enabledClasses} \land \\
\quad\quad f.\textit{name} \in \mathrm{dom}\ m.\textit{consumes} \land \\
\quad\quad \textit{unsatisfied} \in m.\textit{consumes}\,(f.\textit{name}) \land \\
\quad\quad s \in f.\textit{methods} \land \\
\quad\quad s.\textit{name} = \textit{unsatisfied} \land \\
\quad\quad s.\textit{optional} = \textit{NO} \land \\
\quad\quad (\neg\ \exists\, m' : \mathrm{ran}\ \textit{modules};\ \textit{provider} : \textit{ClassName} \bullet \\
\quad\quad\quad m'.\textit{enabled} = \textit{YES} \land \\
\quad\quad\quad \textit{provider} \in \mathrm{dom}\ m'.\textit{produces} \land \\
\quad\quad\quad \textit{provider} \in m'.\textit{enabledClasses} \land \\
\quad\quad\quad m'.\textit{produces}\ \textit{provider} = s.\textit{name} \\
\quad\quad) \\
\quad)\} \\
\hline
\end{array}
$$

If no dependency is missing, the factories and data providers can then be created. When creating the factories and populating them, the Module Manager needs to satisfy a *Setter* method.

$$
\begin{array}{|l}
\hline
\textit{satisfySetterMethod} \\
m, m' : SetterGetter \\
dataProviders? : \mathbb{P}\ ClassName \\
outputid : ClassName \rightarrow DataId \\
\hline
m'.name = m.name \\
m'.argType = m.argType \\
m'.optional = m.optional \\
(\exists\, matching == \{d : dataProviders? \mid outputid\ d = m.name\} \bullet \\
\quad (\#\, matching = 1 \Rightarrow m'.value = (\mu\, m : matching \bullet m)) \land \\
\quad (\#\, matching > 1 \Rightarrow m'.value = dataProviderError) \land \\
\quad (\#\, matching = 0 \land m.optional = YES \Rightarrow m'.value = m.value) \land \\
\quad (\#\, matching = 0 \land m.optional = NO \Rightarrow m'.value = dataProviderError)) \\
\hline
\end{array}
$$

The *satisfySetterMethod* transforms the value of a *SetterGetter* method after having matched an *outputid* to its *dataProvider* from a set of input *dataProviders*.

The whole expression (μ part) returns the unique value of matching, which corresponds to having an *outputid* equal to the name of the *SetterGetter* method with that *outputid* belonging to a set of input *dataProviders*. There are four cases: the sweet path corresponds to having exactly one matching, leading to setting the value of the method to it; another path has no match found but the matching was optional, leading to an unchanged method; and two others where an error is thrown (when no matching is found but it is not optional, and when there are more than 1 match).

$$
\begin{array}{|l}
\hline
\textit{satisfyFactoryInputs} \\
dataProviders? : \mathbb{P}\ ClassName \\
factory?, factory! : Factory \\
\Xi ModuleManager \\
\hline
factory?.name = factory!.name \\
factory!.methods = \mathrm{ran}\,\{m : factory?.methods;\ m' : SetterGetter \mid satisfySetterMethod\} \\
\neg\ \exists\, m : factory!.methods \bullet m.value = dataProviderError \\
\hline
\end{array}
$$

The *satisfyFactoryInputs* modifies an input factory by setting their *dataProviders*. The output methods will be such that each input factory method that is a *SetterGetter* is modified using the *satisfySetterMethod*, if there is no error returned by it.

At this stage the factories have had their data associated to them, and the required *assets* and *behaviours* can now be created. Each individual asset factory has its assets created and added to the *ABMState*, which is the central class for the simulation. Once the network of assets is built, the *BehaviourFactories* call their method to set their behaviours so that the agents are now complete. The ABMState can then call on its scheduler and start the simulation by stepping through the different agents behaviours. All these steps are handled by the *ModuleManager* and do not require the user to enter any code, they just have to specify the required modules and parameters.

5.4 DISCUSSION

This section describes the different challenges and benefits that come from using our automated approach to building large-scale agent-based models. An example is also given that shows that this approach is applicable to other types of problems, broadening its use to applications outside of networked models.

5.4.1 Challenges in building large-scale ABMs in an automated manner

Below are some of the challenges arising from automating building large-scale ABMs.

Bringing different aspects of the model into a networked structure - ordering of the assets' creation

The first challenge comes from building the agent-based model when factories are created by independent authors but there is a need to specify how the agents relate to one another without having a programmer specifying the connections manually.

Figure 5-6 illustrates this issue where two modules have been defined: one containing information about a base network, and the second one about battery technology. The batteries need to be added to that network, either to a premise (at the leaf of the tree), or to any asset on the grid (at nodes within the tree). In order to specify this location, data can be used that will specify what type of battery it is and the system can know what type of network asset it is to be attached to. This can be by referencing the identifier of the network assets, or by following a rule on how to place them. In either case, the batteries rely on the assets defined in the network already being created, which infers a notion of reference.

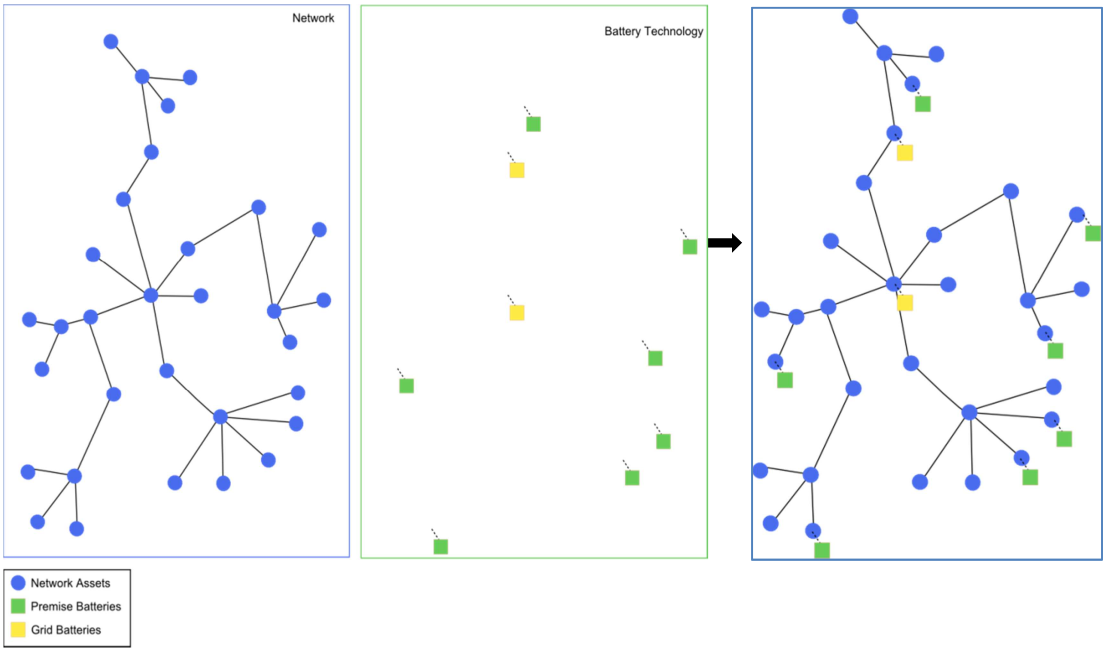


Figure 5-6 - Composition of an agent-based model made of a network and two types of battery technologies which assets have been implemented in separate modules

If the assets are all created within one factory, their ordering can be handled by the developer within that factory. However, if the assets are defined in independently developed factories, there is a need to mention in which order the assets need to be created, and consequently the order in which the factories are being called, so that the Module Manager can automatically handle them. This type of dependency is one of the consequences of aiming at creating an extensible framework, and is the equivalent of inheritance in object programing.

To support this, partial ordering of asset factories has been handled through the schema definition of the plugin extensions. An attribute (*Predecessors*) allows defining which other factories need to be called before this one. The ordering of the asset factories is then managed automatically within the Module Manager, following partial ordering algorithms. Predecessors that are not included in the current model are ignored, so that maximum flexibility is allowed. For example, if factory F has predecessors A and B, it is possible to run models with any combination of F, A and B. If A and F only are part of the model, then F will automatically be run after A, while B is ignored.

Thanks to the separation of the *Assets* and the *Behaviours*, this ordering is only necessary on the assets which describe the underlying network structure of the model and hold the relationships information.

Verification of the models and simulations being created

The second challenge comes from ensuring that code written in independent modules, or plugins is correct, independently but also when they come together at runtime.

Verification of the models is supported through unit testing (Agile Alliance, 2011). Unit tests are performed on every class created within their own plugin, to ensure the correctness of the different operations undertaken in the classes' methods. In some cases, because of some plugin dependencies constraints and the dependency of some factories on readers, dependency injection (Fowler, 2004) is used. This design pattern, which removes hard-coded dependencies at compilation or at runtime, is used to test a factory requiring a data provider for example. While many different data providers could be used, only one needs to be implemented for the test; injected stub dependencies are then used in the plugin where the factory is tested.

System testing is also performed and consists in bringing together different combination of plugins to set up a simulation and run it. Pairwise testing (Czerwonka, 2014) was undertaken to ensure that all possible combinations of pairs of plugins have been tested; the 'Jenny' tool was used here.

Validation of the models

The third challenge comes from ensuring that the models created are valid. The validation of the M&S application was performed by comparing output simulations at locations and for times for which actual measures were available and which had not been used as input to the simulation. This gives confidence in the validity of the behavioural or representational accuracy of the simulated system. As an example of this, using input data at the finest level of granularity (the premise level) and running a simulation, the outputs of the simulation were compared at the feeder level with actual data that had been recorded over the simulation period.

This was done for a base set of simulations. However, because of the vast range of options in building simulations and because the available modules continually expand, it is then the responsibility of the user to check they are valid. It can however be envisaged that some parts of the validation process could be automated, which is part of the future work.

5.4.2 Benefits in building large-scale ABMs in an automated manner

The automated approach described in this paper was chosen for reasons expressed in section 5.2. In addition to the properties of flexibility and extensibility in the model building that were initially sought, other benefits have been identified. These are described here.

Rapid model building

As the model grows and becomes more sophisticated, the creation of new agents requires creating quite a few factories and data provider classes, and links between the plugins that host them. It can be argued that these steps require quite a lot of time to implement, slowing down the overall development progress. To this end, we measured speed in building the model.

A new plugin was recently added to model electric vehicles. This plugin contains an asset factory that describes the characteristics of the vehicles with the point of view of the electricity network; that is they have the characteristics of a battery, with the additional properties of mobility. Another plugin hosted the behaviour factory that describes how the vehicles will behave at each time step, which is dependent on the driving characteristics of the user (time of arrival at the charging station, and distance travelled for the day) as well as the characteristics of the charging method of the battery (whether it can charge at any time of the day or only according to pre-set charging times set from demand management policies). In addition to these, data providers were implemented to specify the characteristics of the vehicles (battery capacity, depth of discharge, efficiency, inverter…) and which assets they are attached to.

The extension of the model with this addition took an experienced programmer a day to implement the different factories and readers and their associated tests. The implementation of the electric vehicle behaviour took the longest in this exercise, as it describes the different charging strategies and the charging requirements. Creating and linking the factories to allow the Module Manager's automation took approximately one hour.

While this approach does not overly extend the time in building the model, it has the advantage of allowing implementations in parallel by different programmers. Indeed, a programmer could implement the asset factory while the other implements the behaviours logic or the data providers. This can considerably speed up the model building time.

Vast number of scenarios can be tried

Thanks to the automation of the model building, it is quite simple for a user, who does not need to program, to create a large number of scenarios. Using the command-line scripts, a user can easily specify which modules and factories to be used to define a specific agent type.

In addition, variations of an initial scenario can easily be tried where for example, one behaviour type can be changed by calling on a different factory in the command line. This would allow comparing easily the impact of some aspects of the agents over the whole system by simply swapping them. It is in a way the equivalent of parameter sweeping but with behaviours defined in factories instead. Parameter sweeping is also possible here as parameters can be specified in the scripts. Further, data used to populate the model can also be modified by simply changing the path of a file to refer it to another one.

By simply specifying these parameters in the command scripts, many alternative scenarios can be set up and the Module Manager will deal with them to create the ABM.

Reproducibility of the simulations is facilitated

In addition to the vast number of possible scenario, reproducibility of simulations is facilitated by the use of the command scripts, in which a random seed is specified. Thanks to the configuration files in which the command scripts are saved, it is possible to rerun a simulation previously done at any time. A random seed was set at the global level within the *ABMState* to ensure that all the actions in the automations are repeated in the same order each time a simulation is run with the same seed. This is so that the plugins and factories specified in the scripts can be discovered and loaded in any particular order, apart from the asset factories on which precedence has been specified. Further, all uses of hashmaps in the code were replaced by LinkedHashmap to enforce maintaining the order. Thanks to these measures, simulations can be repeated with confidence using the same scripts, as the simulation output is deterministic so consistent results will be obtained. This does not prevent the agents to evolve independently; it just guarantees reproducibility of the simulations while still having randomness in their behaviour execution.

5.4.3 Application of MODAM to other problems

Throughout this paper, the automation of building large-scale agent-based models based on a dynamic agent composition approach was argued for the particular application of electricity distribution networks. Here, we are discussing an example that was implemented to test the flexibility of the MODAM framework, using a very different model to those for which MODAM was built, and to test the speed and ease in building the model and running the simulation. The example chosen is an example code (http://jade.tilab.com/doc/examples/party.html http://jade.tilab.com/) distributed with the Jade software.

This example, called "Party" describes a rumour spreading within a crowd of guests at a party. The algorithm was implemented in MODAM to test the flexibility of the platform, and test that automation of building the agent-based model was still possible, as well as to compare speed in simulation execution. This example is most suited for event-driven simulations, as message passing is the method used by the agents to communicate with one another. MODAM, which is best suited for discrete time simulations using time-stepped mechanisms, could however be used, adapting the implementation to provide a similar logic of spreading the rumour.

The example was implemented in one plugin that contains three factories: a *GuestBehaviourFactory*, extending *IBehaviourFactory* that creates two types of guest behaviours (an introBehaviour and a replyBehaviour), a HostFactory, extending *IBehaviourFactory*, that creates one type of behaviour (a hostBehaviour), and a PartyFactory, extending *IAssetFactory* that creates the connections between the different guests and the host. No *DataProvider* was created as this simulation does not require external input data.

The implementation for this simple problem took three hours for 2 people working in a peer-programing environment. This is a reasonable length of time, especially as this implementation required adaptation because of the paradigm differences between the rumour problem and the software purpose.

Simulations were run with an increasing number of agents to test the simulation speed. This was done both on the MODAM and the Jade implementation, whose speed results are given in Figure 5-7. The number of agents presented here is limited to 1000, as the Jade GUI does not allow larger number of agents; simulations with MODAM could be run with as many agents as wanted.

		Simulation Speed (in s)	
		Jade	MODAM
Number of Agents	10	0.042	0.0022
	100	0.271	0.039
	500	1.542	0.1734
	1000	3.206	0.5264

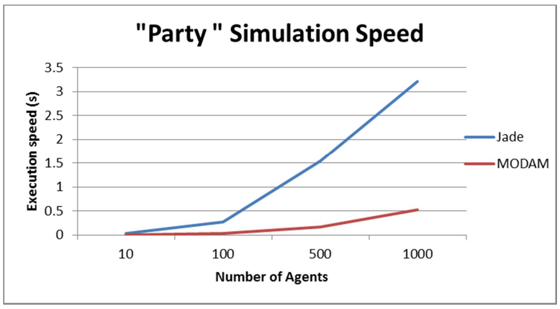


Figure 5-7 - Simulation speed results for the "party" implementation with Jade and MODAM.

We can see in Figure 5-7 that the MODAM implementation was faster than the Jade implementation up to 1000 agents, despite it not being developed for this type of problem. While we do not know how they compare for larger numbers of agents, MODAM simulations speed started to decrease as the number of agents reached 100,000; however, its agents' steps per second remained constant. This is due to the fact that every single agent will be stepped at every timestep; while if using a message-passing method, the number of steps by the agents would be less, leading to a speed increasing only slightly as the number of agents increases. While MODAM is not built for this type of paradigm and best suited for simulations over networked structures, it could however be adapted.

5.5 RELATED WORK

Agent-based modelling has been used successfully over the last decade to model different aspects of the electricity sector. Their use was initially mainly for the analysis of power market design for large-scale electricity systems when deregulation happened (Batten & Grozev, 2006; North et al., 2002; Weidlich, 2008). These models aimed at investigating the interactions between the physical infrastructure at the transmission level, and the economic behaviour of market participants to help engineer markets in the electricity sector. Infrastructure transitions of the electricity sector has also been assessed using ABM (Chappin & Dijkema, 2010) with models of the impact of CO_2 policy on the power production sector, or the transition to global liquefied natural gas infrastructure. More recently, ABM has been applied to the demand response management area where modelling of supply and demand was done in the view to balance the conflicting interests of the different supply and demand actors as well as the distribution network operators (Boait et al., 2013).

The application of ABM to the electricity distribution network (medium and low voltage networks) is not as widespread as that of the transmission networks (high voltage). One power system modelling and simulation environment however, that can describe the infrastructure at a lower level, is GridLab-D (Chassin et al., 2014; Chassin et al., 2008). GridLab-D describes the different elements that compose the distribution grid up to the transmission grid point(s), taking into account the technical characteristics of the system; that is, describing the physical components as well as their behavioural interactions. Simulations can be performed directly within the platform, but it has also been used in combination with other systems. For example, (Cai et al., 2011) discusses a model of the impact of high penetration of photovoltaic generation on the market price, where the characteristics of the distribution network were taken into account. More recently, the European Commission has published some work (Institute for Energy and Transport, 2014) showing the integration of the physical infrastructure of a distribution network with the interaction of prosumers; however, no detail is given in the way the agents and the infrastructure are described. Their agent-based models were created to answer specific questions, where parameters can be varied over many simulations, but which model needs to be modified programmatically if new questions are to be answered. These models get created within a central location, and lack the extensibility and flexibility characteristics required by the constraints of our project.

Other software applications, also called agent-based or multi-agent can be found in the literature; however, they might refer to other notions depending on the context (van Dam et al., 2012). The definition of an agent chosen in the context of this paper is the one defined in (Balci, 2013) where an agent is "*'intelligent',* *adaptive, autonomous, goal/self-directed, has the ability to learn, and can change its behaviors based on experience".*

The definition of an agent in software engineering is similar; however, it does not describe the same type of environment. An agent might be a piece of code that can run independently on different machines for example. The code encapsulated in an agent is similar to that of an object in object-oriented programing, except that an agent can decide whether it will carry out a request while an object responds to a request. In this case, the whole software system is composed of many agents that can communicate with one another to perform different functions of the software without the need for the classes to represent themselves but rather objects that will carry out requests. There can therefore be many applications to that type of software architecture. These types of software architecture are also very popular for distributed systems where the different functions as described into agents can be executed on different computers or cores. An example of such an architecture is Cougaar which is a Java-based architecture for the construction of large-scale distributed agent-based application (Helsinger, Wright, & Ieee, 2005). Another application is EPOCHS (Hopkinson et al.) which is a federated simulation system that uses the term agent to refer to computer programs that communicate with one another. The agents in our agent-based model are the finest level of granularity of the system, and contain the information describing their behaviour through their implementation; an EPOCHS 'agent would be the piece of software that could encapsulate many agents for example. This is also something that we have been using and that we call a module, or component, in this paper.

As can be seen here, the term *agent* has been used in the two domains of modelling and software engineering, in two different contexts, while they have a lot in common. In ABM, the model can also be implemented using different software architectures that have been described by North and Macal ((North & Macal, 2007), chapter 7), as tightly coupled, loosely coupled and distributed architectures. The references to these architectures mainly concern the way the different functionalities are implemented – that is the level of coupling between the user interface, the simulation engine and the data storage. In both cases of loosely coupled and distributed architecture however, the ABM model which is run in the simulation engine is still implemented in a central place where the relationship between the agents or groups of agents are manually implemented.

The work presented in this paper goes one step further, where the model itself is built in compositional manner, where an agent can be implemented in a separate module independently; it will then be integrated with other agents to make the model at simulation initialisation in an automated manner. The implementation follows the principles of information hiding (Parnas, 1972), where separation of concerns (Dijkstra, 1982) is respected not only at the object level (as in the object-oriented paradigm) but also at the software level, as in component-based software engineering (Szyperski, 1997). Component-based software emphasizes reuse as components can be saved independently and assembled to create new software systems, saving development time and cost. They are also more flexible and scalable which is a feature sought after in this simulation environment implementation.

From these components, composability which is "*the degree to which an artefact is capable of being constituted by combining things, parts, or elements*" (Balci, Arthur, & Ormsby, 2011) is then obtained. Composability has been used in the electricity sector, with for example an application in the area of smart grids where different simulators are brought together so that large simulations can be run (Schutte & Sonnenschein, 2012). The aim in our project was not however to bring existing simulators together but rather using similar principles of composability and reusability to easily extend an agent-based model, and create large-scale scenarios.

In a way, the environment presented in this paper combines the two approaches presented above. The agents can be defined in separate modules, and get weaved together at simulation setup to create the full agent-based model on which simulations can be run. The agents then run autonomously in the simulation environment, leading to the same results as would be obtained if the agent-based model was built by defining the connections between the agents in a central place. The advantage of our approach is that extending the model through adding or modifying the agents and their behaviours, is facilitated by this component approach. The ABM is thus built weaving together modules of the different parts making up the model in an automated manner, which is novel.

5.6 CONCLUSION

This paper has presented how large-scale agent-based models can be built in a flexible and extensible manner so that simulations can easily be set up. For this, an automated approach to building the agent-based model was chosen and presented in this paper. Explanations about three aspects that support this automated process were given: the technology used to implement the system, the way large-scale ABMs are specified using command scripts or GUIs, and the automation of the composition by formally specifying the Module Manager, using Z.

While answering the project requirement and providing additional benefits in taking such an approach, this method has shown challenges which have also been described here, along with our answer to them. Finally, while developed specifically to our research problem, MODAM can be used for applications other than networked structures as demonstrated in an example. While applicable to other problem structures, MODAM is best suited to networked structures that use a time step paradigm.

Future work includes applying MODAM to create many scenarios using models of uptake of different technologies developed by other agencies to examine the impact they will have on a specific electricity grid. It is also planned to extend the model further with additional agents including demand management. MODAM can be downloaded for interested users.

5.7 ACKNOWLEDGEMENTS

The authors gratefully acknowledge the funding through the NIRAP grant, which is making this research possible, the contributions of diverse partners on this project and especially Ergon Energy for providing the data used in the framework.

5.8 REFERENCES

AEMO. (2012). Rooftop PV Information Paper - National Electricity Forecasting (pp. 60): AEMO - Australian Energy Market Operator.

Agile Alliance. (2011). Unit Testing. Guide to Agile Practices. Retrieved 05/10/2014, 2014, from http://guide.agilealliance.org/guide/unittest.html

Australian PV Institute, & Australian Renewable Energy Agency. (2013). Australian PV market since April 2001. Retrieved 08/01/2014, 2014, from http://pv-map.apvi.org.au/analyses

Balci, O. (2013). Introduction to Modeling and Simulation. ACM Special Interest Group on Simulation and Modeling (SIGSIM) Modeling and Simulation Knowledge Repository (MSKR). http://www.acm-sigsim-mskr.org/Courseware/Balci/introToMS.htm

Balci, O., Arthur, J. D., & Ormsby, W. F. (2011). Achieving reusability and composability with a simulation conceptual model. Journal of Simulation, 5(3), 157-165. doi: 10.1057/jos.2011.7

Batten, D. F., & Grozev, G. (2006). NEMSIM: Finding Ways to Reduce Greenhouse Gas Emissions Using Multi-Agent Electricity Modelling Complex science for a complex world: exploring human ecosystems with agents (pp. 227-252). Canberra: ANU E Press.

Boait, P. J., Ardestani, B. M., Mark Rylatt, R., & Richard Snape, J. (2013). Managing complexity in the smart grid through a new approach to demand response. Emergence: Complexity and Organization, 15(2), 23-37.

Cai, C., Jahangiri, P., Thomas, A. G., Zhao, H., Aliprantis, D. C., & Tesfatsion, L. (2011, 24-29/07/2011). Agent-Based Simulation of Distribution Systems with High Penetration of Photovoltaic Generation, San Diego, CA.

Castiglione, F. (2006). Agent based modeling. Scholarpedia, 1(10), 1562. doi: doi::10.4249/scholarpedia.1562

Chappin, E. J. L., & Dijkema, G. P. J. (2010). Agent-based modelling of energy infrastructure transitions. International Journal of Critical Infrastructures, 6(2), 106-130. doi: 10.1504/ijcis.2010.031070

Chassin, D. P., Fuller, J. C., & Djilali, N. (2014). GridLAB-D: An agent-based simulation framework for smart grids. Applied Mathematics(Journal Article).

Chassin, D. P., Schneider, K., & Gerkensmeyer, C. (2008). GridLAB-D: An open-source power systems modeling and simulation environment. Paper presented at the Transmission and Distribution Conference and Exposition.

CSIRO Future Grid Forum. (2013). Change and choice: CSIRO.

Czerwonka, J. (2014). Pairwise Testing. Retrieved 04/05/2014, 2014, from http://www.pairwise.org/tools.asp

Dijkstra, E. W. (1982). EWD 447: On the role of scientific thought. Selected Writings on Computing: A Personal Perspective, 60-66. doi: citeulike-article-id:2490230

Ergon Energy. (2013). Corporate profile. Retrieved 02/06/2013, 2013, from http://www.ergon.com.au/about-us/company-information/corporate-profile

Fowler, M. (2004, 23/01/2004). Inversion of Control Containers and the Dependency Injection pattern. Retrieved 05/12/2013, 2013, from http://martinfowler.com/articles/injection.html

Hayes, I., Flinn, B., Gimson, R., King, S., Morgan, C., Sorensen, I. H., & Sufrin, B. (1986). Specification Case Studies (I. Hayes Ed. 2nd ed.): Pearson Education Limited.

Helsinger, A., Wright, T., & Ieee. (2005). Cougaar: A robust configurable multi agent platform 2005 IEEE Aerospace Conference, Vols 1-4 (pp. 3129-3138).

Hopkinson, K., Xiaoru, W., Giovanini, R., Thorp, J., Birman, K., & Coury, D. EPOCHS: a Platform for Agent-Based Electric Power and Communication Simulation Built from Commercial Off-the-shelf Components. 21, 548-558. doi: 10.1109/tpwrs.2006.873129

Institute for Energy and Transport. (2014, 18/07/2014). Agent Based Modelling for Smart Grids. Retrieved 20/07/2014, 2014, from http://ses.jrc.ec.europa.eu/agent-based-modelling-smart-grids

Klügl, F., & Bazzan, A. L. C. (2012). Agent-based modeling and simulation. AI Magazine, 33(3), 29-40.

North, M., Conzelmann, G., Koritarov, V., Macal, C., Thimmapuram, P., & Veselka, T. (2002). E-laboratories : agent-based modeling of electricity markets. Paper presented at the American Power Conference, Chicago, IL (US). http://qut.summon.serialssolutions.com/2.0.0/link/0/eLvHCXMwY2BQMEhMMjRMTDUzTDKwTDQzSkwzTQJGfaqlaUpyslEq5BBt-GAbUmnuJsTAlJonyqDu5hri7KGbD0zj8QWQMxfiQacggwVAByCbW4Im3wzFGHgTQcvA80rA28VSJBgUEg2TLJIsks2MDJLNTVKA3TszM_PU5FSDVNMk4zQTgyQAauMqgA

North, M. J. (2013). A theoretical formalism for analyzing agent-based models. Complex Adaptive Systems Modeling, 2(1), 3-3. doi: 10.1186/2194-3206-2-3

North, M. J., & Macal, C. M. (2007). Managing Business Complexity: Oxford University Press.

OSGI Alliance. (2013). OSGI Alliance. Retrieved 01/02/2013, 2013, from http://www.osgi.org/Main/HomePage

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. Communications of the ACM, 15(12), 1053-1058. doi: 10.1145/361598.361623

Schutte, S., & Sonnenschein, M. (2012, 2012). Mosaik - Scalable Smart Grid scenario specification. Paper presented at the Winter Simulation Conference, Berlin, Germany.

Szyperski, C. (1997). Component software: beyond object-oriented programming. New York: ACM Press.

van Dam, K. H., Nikolic, I., & Lukszo, Z. (2012). Agent-Based Modelling of Socio-Technical Systems. Dordrecht: Springer Netherlands.

Vogel, L. (2012, 08/05/2012). OSGi Modularity - Tutorial. Retrieved 24/09/2012, 2012, from http://www.vogella.com/articles/OSGi/article.html

Weidlich, A. (2008). Engineering interrelated electricity markets: an agent-based computational approach. Heidelberg: Springer [distributor].

Chapter 6: Parallel ABM for Electricity Distribution Grids: a Case Study

This chapter has been written as a conference paper[4]. It describes the parallel implementation of the scheduler so that simulations using MODAM can be sped up, which is important especially as the model grows and the number of agents increases greatly. Taking advantage of the structure of the model, which is a directed acyclic graph, the scheduler groups agents and executes them in parallel over different threads. A description of the implementation is given, followed by an example on which speed tests have been performed, showing that a speedup of 2.6 could be obtained on a medium size network using our parallel implementation.

It has to be noted that terminology is different in this paper to the one introduced in the previous two chapters. This paper, which was written first, uses the term agents to reference what we later described as behaviours of agents. Initially, we saw the assets and their relationships forming a graph as the space over which the agents were evolving. However, because of the connections of the assets that are necessary to represent the coupling of the agents, as well as to keep some conformity with most descriptions of agents as having attributes and behaviours (North, 2013), the term agent was replaced by behaviour.

[4] This paper was presented during the 1st Workshop on Parallel and Distributed Agent-Based Simulations as part of the Euro-Par 2013 conference, and was published in volume 8374 of the Lecture Notes in Computer Science series

Parallel ABM for Electricity Distribution Grids: a Case Study

Fanny Boulaire, Mark Utting, Robin Drogemuller
Queensland University of Technology, 2 George Street, Brisbane, Queensland 4000, Australia

Parallel ABM for Electricity Distribution Grids: a Case Study

ABSTRACT

This paper introduces a parallel implementation of an agent-based model applied to electricity distribution grids. A fine-grained shared memory parallel implementation is presented, detailing the way the agents are grouped and executed on a multi-threaded machine, as well as the way the model is built (in a composable manner) which is an aid to the parallelisation. Current results show a medium level speedup of 2.6, but improvements are expected by incorporating newer distributed or parallel ABM schedulers into this implementation. While domain-specific, this parallel algorithm can be applied to similarly structured ABMs (directed acyclic graphs).

Keywords: electricity distribution grid, network, composable ABM.

6.1 INTRODUCTION

Agent-based models (ABMs) have been used for diverse applications in many domains for their ability to capture the actions and interactions of the agents composing a complex system so that the overall behaviour of the system can be assessed (North & Macal, 2007). This paper describes an ABM applied to the electricity domain. The aim of the project this work belongs to is to develop a planning tool for optimal investment strategy of electricity distribution networks over large areas and long planning horizons. A mix of renewable energy, energy storage and other new technologies are compared with traditional practices to assess their impact on the operation of the distribution network.

Traditionally, electricity has been generated at large centralised power stations and then distributed to consumers over a wired network. The introduction of renewable energy systems at various levels within the transmission/distribution system adds complexity to the system due to injection of electrical power in locations that were not initially designed to receive power, mismatches between time of availability and time of demand and reduced control over the phase of generated power. Ergon Energy (Ergon Energy, 2013a), the industry partner in this project, provides electrical power to approximately 700,000 customers over an area exceeding 1,000,000 km^2. The Ergon network consists of approximately 150,000 km of power lines and 1,000,000 power poles, together with associated substations and transformers, and has a total asset value of AU$10.6B. Ergon are using the software presented in this paper to support network infrastructure investment decisions, looking for the most appropriate combination of technologies and practices to reduce overall costs.

An Ergon user interacts with the software by selecting nodes within the network structure to be analysed against various scenarios, observing the impact these changes within the network might have on the overall system. A user may analyse a single network topology against a range of user demand predictions or, inversely model user demand predictions against a range of potential network configurations. Initially the system represented the infrastructure of the distribution grid, the attachment points to the transmission grid, solar and wind renewable energy sources as well as emergency diesel generation and the different types of consumers.

During development it was recognised that battery storage and the demands of electric vehicles will be needed.

This tool is therefore being built to answer a wide range of questions, which can arise as time passes, taking into account both the technical and economic constraints of the system, and doing so in an integrated manner. Consequently, the tool was built in a composable manner to ensure its flexibility and extensibility. The wide range of physical asset types within the system, the wide range of potential configurations within each asset type, the need for communication of deep, complex issues between the software engineers and domain experts and a desire for flexibility into the future all indicated that an ABM approach had desirable characteristics. ABM was chosen as the modelling technique mainly because the evolution of the electricity grid is unknown due to the many new technologies being employed; in this case the past is no predictor of the future. Also, the ability to describe the agents at various scales, both over space and time, was important to capture the different system characteristics accurately and dynamically.

As the model grew, adding more agents of varied types, the simulation slowed down, leading to the need for parallelization. A parallel implementation of the agent scheduling was done to speed up the execution time and because the structure of the model could be broken down into independent tasks with known common patterns. This paper presents this parallel implementation using a fine-grained shared memory parallelism performed on a multi-core computer, detailing the technical implementation dependent on the overall tool architecture.

The first part of this paper describes the overall architecture of the tool, built in a composable manner, which defines the requirements of the implementation. Then, the description of the parallel implementation within that software framework is given and performances of the parallel implementation are discussed.

6.2 ABM PLANNING TOOL ARCHITECTURE

This section briefly describes the overall architecture of the ABM planning tool to provide an understanding of the constraints on the agent-based model, which in turn influences the parallel execution of the agents.

6.2.1 Composition of the model using plugins

Because the tool was built with the need for evolution in mind, so that many questions could be answered and these in an integrated way, a composable approach was taken. For this, OSGi (formerly Open Services Gateway Initiative) (OSGi Alliance, 2013), a specification that enables writing modular software, was chosen as the underlying technology for the software. As described by its community of users, it "*reduces complexity by providing a modular architecture for today's large-scale distributed systems as well as small, embedded applications*". This specification has been used by the Eclipse Community, whose plugins are OSGi bundles (Vogel, 2012). The tool presented in this paper is built using Eclipse plugins which are our unit of modularity.

The breakdown of the software in plugins is done over different layers. First, one main plugin is defined, called MODAM (MODular Agent Model), which can be seen as the core of the whole framework. It contains the schemas that define the different extensions, which are used by the different plugins to support interconnection (Blewitt, 2007). It also contains the schedulers (sequential and parallel), and ensures the automatic creation of the objects and agents from the plugins using factories.

The different plugins contributing to the creation of the ABM are defined according to their logical meaning. A plugin can be defined for a given analysis type, for example the amount of electricity a solar panel can produce. Within each of these logical units further breakdowns can be performed, to contain different logical or functional units. So far, 3 types of functional units have been distinguished within a logical one. These can be seen through the different extensions that are defined in the MODAM plugin: asset factory, agent factory and data provider. Details about the software architecture can be found in (Boulaire et al., 2013a). Using solar panels as an example, the asset factory creates the solar panel assets with their physical characteristics (capacity, derate factor…), the agent factory the behaviour of the solar panel according to a photovoltaic (PV) output model dependent on the weather conditions, and the data provider reading the weather information at the location of the solar panel, being an input to the PV model.

A distinction has been made between assets and agents; this is further explained below.

6.2.2 Integration of assets and agents over separate layers

When looking at an implementation of an agent in an ABM, it is normally mainly defined through its behaviour in relation to its environment and other agents, using only the characteristics that are required to determine the agent's action. That is, the object's characteristics and behaviours are combined in one single class and only the necessary attributes are modelled. Also, the agents are often instantiated in a central class where the relationships amongst given agents are set, as well as their scheduling for the simulation. Such an approach can be sufficient when building a model that has well-defined boundaries. However, when planning on extending the functionality of a model, such an approach can be restraining, as reuse is limited because of this tight coupling and access to the code is required to extend the model.

For our ABM implementation, a distinction has been made between the characteristics and the actions of our agents. Figure 6-1 shows a schematic representation of the two distinct layers that compose our model implementation: the asset and the agent layer. In the context of this paper, an 'asset' is the object that contains the physical aspects, and an 'agent' holds only the behaviour. The two are however linked, and the behaviours can be informed using the asset's characteristics.

Figure 6-1 shows that the different elements composing the simulation are coming from different factories that are in separate plugins. Often there is a one-to-one relationship between an asset and an agent, but it is not required. Indeed, as shown on the figure, an agent representing the behaviour of a single asset can be built combining multiple behaviours – this is the case for a premise agent, which is the combination of electricity consumption and demand-side management behaviours which can come from two distinct plugins and modified as more information becomes available to define the behaviour. It is not necessary to load both behaviours for all scenarios. This architecture also supports the loading of alternative behaviours, for example consumption, against a single implementation of demand-side management. Conversely, a single behaviour can be assigned to a group of assets, which is the case when doing a load flow analysis for example.

Finally, it can be seen that while many plugins have been used to implement the model, only one integrated network graph has been created and it runs as one single agent-based model – the one on the Agent Layer.

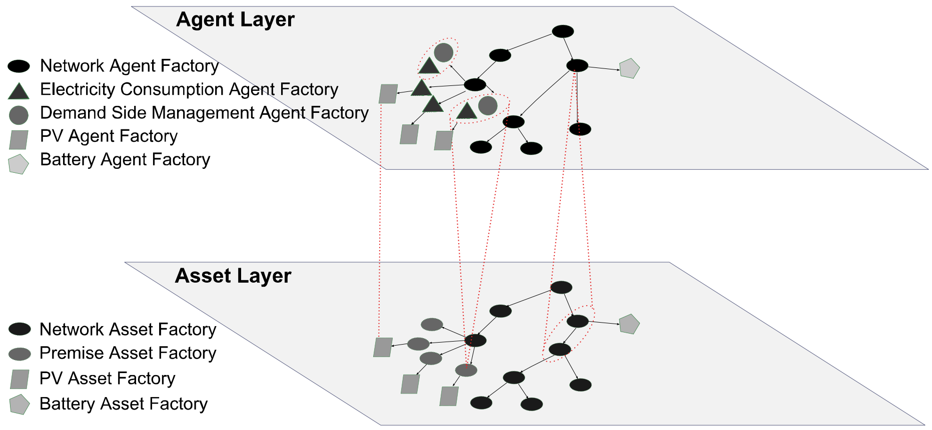


Figure 6-1 - Schematic representation of the ABM architecture.

6.3 PARALLEL ABM FOR THE ELECTRICITY DISTRIBUTION GRID

Having defined the structure of the agent-based model implementation, this section describes its parallel implementation, which is a fine-grained parallelism on a single shared memory computer. The following sections describe the different aspects relating to the parallelization of the agent-based model execution:

- Ordering of the agents at simulation setup
- Parallel execution of the agents at runtime

It should be noted that the application presented in this paper is not a generic solution for parallel ABM; it is related to the specific case study presented here. However, because the nature of the problem is a graph structure, whose execution benefits from parallelisation, it can be applied to other applications of similar nature. Also, a distributed ABM could be built on top of this implementation, but this is outside of the scope of this paper.

6.3.1 Ordering of the agents at simulation setup

Because of the dependence of some agents on other agents that define their relationships and interactions, the ordering of the agents' execution is extremely important. Due to the way the agent-based model is built, this ordering needs to follow the architecture constraints. This means that two types of ordering needed to be considered: ordering within the same plugin, and ordering across plugins.

Because plugins can be used independently of one other, it was necessary to have agents ordered within the plugin in which they are defined. This ordering is handled within the code where the developer has access to the agents themselves through their relationships to one another. Because it is a graph structure, this was handled using '*Before*' and '*After*' attributes, to define a strict *partial order* over the agents. A partial order is a transitive relation that is irreflexive and antisymmetric, i.e. corresponds to a directed acyclic graph which is what is represented in our system.

The second type of ordering regards those agents that are dependent on other agents defined in another plugin. This cannot be defined in advance in either plugin, since it is possible for the plugins to be developed independently so neither one knows about the other. This type of ordering is a bit more sophisticated as it requires ordering agents from at least 2 sets, where within each of these sets, agents are also ordered. For this, partial ordering of the agents is again used. An example of this is given in Figure 6-2. It shows 3 plugins, 2 of which can be run in parallel (Plugin A and B) with the third one requiring having its agents called after both of them. In each of them we have 3 agents; plugin A and C have their agents ordered sequentially, and plugin B has 2 agents that are ordered sequentially (b_1 and b_2) and one that can be called anytime (b_3). The ordering between plugin B and plugin C means that all agents in plugin B must finish their step methods before any agents in plugin C can start.

The ordering of the agents is set through an argument in the command line, using "-order" and a combination of the plugins. In the example given in Figure 6-2, the ordering argument looks like:

-order = (*td*(PluginA) || *bu*(PluginB)); PluginC

Where *bu* stands for bottom up, and *td* for top down ordering; || shows that PluginA and PluginB are to be ordered in parallel, and the semi-colon (;) is to show sequential ordering.

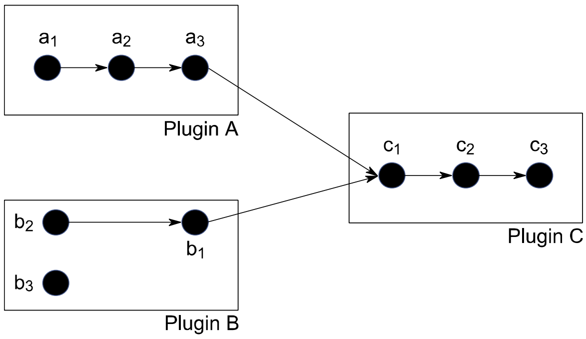


Figure 6-2 - Example of ordering of agents within plugins and amongst plugins.

6.3.2 Parallel execution of the agents at runtime

Given the nature of the problem, each agent in the graph has to wait for one or more agents to be executed before it can proceed with its own action. A barrier to efficient and speedy processing is therefore a blocking condition where an agent is waiting on others' execution. Indeed, early versions of our parallel ABM treated each agent as a separate task, but profiling showed that this resulted in too much overhead synchronising between hundreds of thousands of small tasks.

So, in order to speed up the execution of the ABM, the partial order from the previous section is broken down into many small independent groups on which the calculations are performed. The size of these groups can be varied by the user using the parameter *–ThreadGroupSize* on the command line. Then these groups of agents can be executed in parallel on *n* threads, specified by the Java command line argument *–Threads*. Details of these two phases are as follows:

1) **Setting up of the queue** - Grouping is done at the beginning of a simulation, and whenever new agents are created, by traversing the partial order, using a depth-first search. These groups are defined and queued in a linear order, with each group recording any previous groups that it depends upon. Mathematically, the groups form equivalence classes over the agents and the partial order over the groups is derived from, or is an abstraction of, the partial order over the agents. These group sizes are less than or equal to the number specified by *ThreadGroupSize*, depending on the configuration of the graph – e.g. when an agent has multiple parents.

2) **Execution of the agents –** Execution is done repeatedly for each time step. Within each time step, it is also repeated until the whole graph has been executed.

   a) Depending on the number of threads (*n*), the *n* first available groups of agents, or task, in the pool are executed in parallel. The calculations are performed within each of these groups and assigned to the top node that can be used in the next task.

   b) As soon as a thread becomes free, it accesses the next available task in the pool and checks its dependencies to see if they have finished processing. If they have, the step method of the group of agents is executed.

Figure 6-3 illustrates this algorithm using a small extract of a network with two main branches – one starting with a feeder line L1 and another one with the feeder line L2, that then join at a bus B4. Under that bus, 5 houses are represented, some of which have solar panels installed.

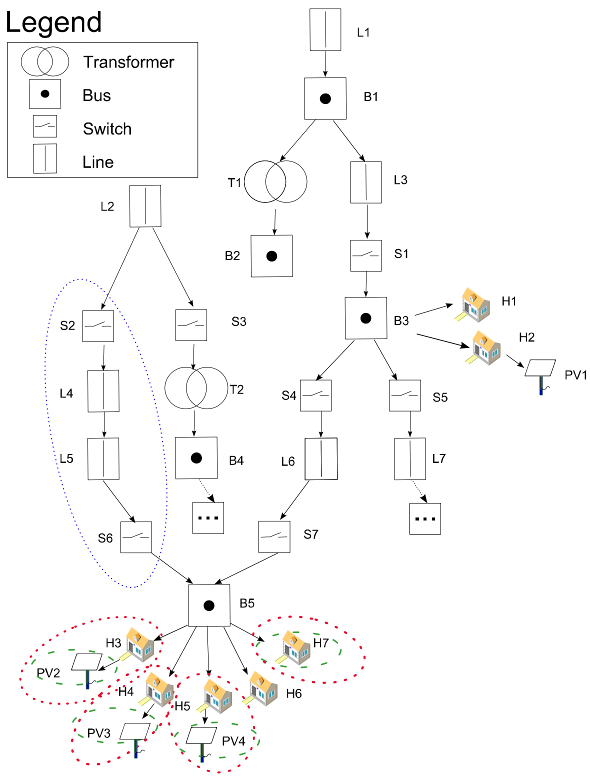


Figure 6-3 - Small extract of a distribution network, implemented with a parallel ABM.

Following the description above, each agent would have been ordered during the simulation setup in the module manager, indicating the order in which the agents need to be run. This ordering can be either top-down or bottom-up depending on the user's instructions.

From this ordering, the pool would have been created, first placing the agents at the leaves of the branches, i.e. those that do not have any dependency on other agents. From the example given in Figure 6-3, with a *ThreadGroupSize* of 1, agents PV1, PV2, PV3, PV4, H1, H6 and H7 would be at the beginning of the queue followed by H2, H3, H4, H5, and so on. With a *ThreadGroupSize* of 4, we would have at the top of the pool, [PV2, H3], [PV3, H4], [PV4, H5], H6 and H7, followed by B5, and then [S6, L5, L4, S2] and [S7, L6, S4]. These groups contain fewer agents than 4 because they all impact on B5 which has multiple parents.

Following this, depending on the number of threads (*n*), the *n* agents are taken from the queue and their step method executed. As the agents' execution is finished, the thread is freed and the following task within the pool is then executed.

6.4 RESULTS AND DISCUSSION

Following the parallelised implementation described above, simulations were performed on an i7-2600 CPU (4 cores + hyper threading) machine. The test case was for a medium-size town in Queensland, containing 75,910 assets of different natures (premises, lines, transformers…), and each asset had exactly one agent in this example. Parameter sweeps were performed and showed that using *n+1* threads (where *n* is number of cores) and *ThreadGroupSize*=1000 would give the fastest runs. Using group sizes of 1000, Figure 6-4 shows the relative speed up obtained for numbers of threads varying from 1 to 10. The *ideal* (dashed) line shows the ideal performance if each of the available processors were fully utilised; from 4 to 8 cores, the slope is lower because hyper-threading typically gives less speedup than real hardware cores (Casey, 2011). It can be seen here that a speed up of 2.6 could be attained when using 8 threads (green line). It can be noted on this graph that there are 3 additional series titled *start*, *step* and *stop*. These correspond to the three phases that have been distinguished within a step method of an agent execution; *step* corresponds to the calculations that are performed after obtaining the information (*start*) that is required to perform its decision and before it is sent (*stop*) to the next agent or stored in memory when requested by an agent. The *step* series shows the most improvement, which is the most important as it requires the most computing.

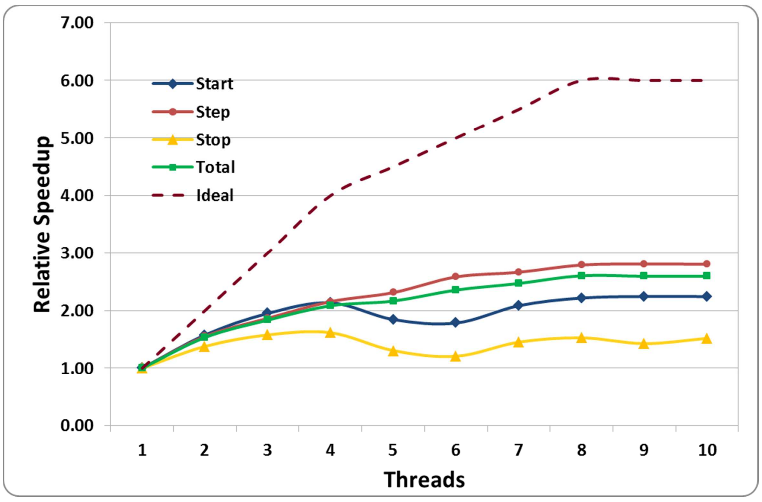


Figure 6-4 - Relative speedup of the parallel implementation of ABM model for the electricity grid.

The speedup obtained here is application-specific, as it depends on the relations between the agents and the granularity of their step methods. Our current scheduling algorithm, which is an NP-hard scheduling problem, gives medium speedup levels. It is however an improvement from other tested algorithms. There is on-going research to further improve the speedup.

An important point is the deterministic execution of the code. The outputs of the simulation are identical whether the simulation is run in parallel or sequentially which is an important feature. In some other implementations of parallel ABMs, determinism is not required, and even sometimes not desirable. In this case determinism is essential because of the impact of order of agent execution, which can be defined as a directed acyclic graph.

The i7-2600 CPU with 4 cores and hyper threading is being used to test the parallel algorithms due to the ability to constrain the entire execution environment. The full system is run on an SGI Altix XE Cluster with 128 nodes and 1924 x 64bit Intel Xeon Cores with a maximum 101.8 TeraFlops using single precision mode and 15,264 GigaBytes (GB) of main memory. Unfortunately, exclusive access to the cluster for experimentation purposes is not possible.

6.5 RELATED WORK

Many applications are available to support the development of agent-based models ((Berryman, 2008); (Railsback et al., 2006)). More recently, some of them have been further developed to support parallel and distributed ABMs, as the need for faster execution times and support of larger and larger models has arisen. This is the case for D-MASON (Cordasco et al., 2013), which is the parallel and distributed version of MASON, or Repast-HPC (High Performance Computing) (Collier, 2013) for Repast Simphony. Other platforms such as JADE (Bellifemine et al., 2003) which is a middleware for distributed multi-agent application based on peer-to-peer communication architecture are also available. Most research on parallelizing ABMs has focussed on distributed computers, but there has been some research on multicore implementations, such as the study of spatial interactions by Gong et al. (Gong, Tang, Bennett, & Thill, 2013). They achieve high speedups from 1-32 cores, but do not ensure determinism, so all their agents can be executed in parallel (with a small amount of locking required to ensure data integrity), which is essentially an embarrassingly parallel scenario. In contrast, we preserve determinism and our applications typically have complex and deep dependency graphs between agents that constrain the possible parallelism.

MASON was initially selected as the ABM engine for our software platform for its ease in separating the engine (which we were interested in) from the rest of the platform as well as for its execution speed. However, it was later replaced by our own implementation of a scheduler in Java. At the time the tool was being developed, D-MASON was not available and its availability has only recently come to the knowledge of the authors of this paper. Consequently, the work presented here has not been tried using D-MASON. D-MASON implements distributed ABMs rather than the shared memory parallelism discussed herein. This fine-grained parallelism is designed to speed up the simulation of networks of closely connected agents with frequent communication, while distributed ABM approaches are more suited for loosely connected sets of agents with less communication. Distributed ABM is complementary to the methods presented here. A combined approach will be tested in the future.

6.6 CONCLUSION

A parallel implementation of an agent-based model for planning the electricity grid has been presented. This implementation, which is application-specific, was based on the structure of the agent-based model, reduced to a directed acyclic graph.

Agents were ordered, and then grouped respecting the ordering, to be executed in parallel over different threads. Results showed a relative speedup of 2.6, using an i7-2600 CPU (4 cores + hyper threading) machine, which further research will aim at improving. While this fine-grained shared memory parallelism is specific to the structure of this ABM, it can be applied to other ABMs with similar agent structures.

This application was built using the MODAM framework (Boulaire et al., 2013a) which aims to answer diverse questions applied to the electricity grid. This software framework, which supports flexible and extensible models, was presented to provide an understanding of the structure of the model. This framework has two schedulers (sequential and parallel) which have been implemented in-house. The use of other schedulers from distributed and parallel ABM tools, such as D-MASON and Repast-HPC, will be investigated in future research to see if the speed of the simulations can be further improved.

6.7 REFERENCES

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2003). JADE - A White Paper. exp in search of innovation, 3(3)

Berryman, M. (2008). Review of Software Platforms for Agent Based Models. (DSTO-GD-0532).

Blewitt, A. (2007). Getting started with Eclipse plug-ins: creating extension points Retrieved 13/02/213, 2013, from http://www.eclipsezone.com/eclipse/forums/t97608.rhtml

Boulaire, F. A., Utting, M., & Drogemuller, R. (2013, 18-26 May 2013). MODAM: A MODular Agent-based Modelling Framework. Paper presented at the 2nd International Workshop on Software Engineering Challenges for the Smart Grid as part of 35th International Conference on Software Engineering (ICSE 2013), San Fransisco, CA, USA.

Casey, S. D. (2011). How to Determine the Effectiveness of Hyper-Threading Technology with an Application. Intel® Developer Zone Retrieved 26/07/2013, 2013, from http://software.intel.com/en-us/articles/how-to-determine-the-effectiveness-of-hyper-threading-technology-with-an-application/

Collier, N. (2013). Repast HPC Manual (pp. 43).

Cordasco, G., Chiara, R. D., Raia, F., Scarano, V., Spagnuolo, C., & Vicidomini, L. (2013). Designing computational steering facilities for distributed agent based simulations. Paper presented at the Proceedings of the 2013 ACM SIGSIM conference on Principles of advanced discrete simulation, Montreal, Quebec, Canada.

Ergon Energy. (2013). Corporate profile Retrieved 02/06/2013, 2013, from http://www.ergon.com.au/about-us/company-information/corporate-profile

Gong, Z. Y., Tang, W. W., Bennett, D. A., & Thill, J. C. (2013). Parallel agent-based simulation of individual-level spatial interactions within a multicore computing environment. International Journal of Geographical Information Science, 27(6), 1152-1170. doi: 10.1080/13658816.2012.741240

Jackson, S. K., Railsback, S. F., & Lytinen, S. L. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. Simulation, 82(9), 609-623.

North, M. J., & Macal, C. M. (2007). Managing Business Complexity: Oxford University Press.

OSGI Alliance. (2013). OSGI Alliance Retrieved 01/02/2013, 2013, from http://www.osgi.org/Main/HomePage

Vogel, L. (2012, 08/05/2012). OSGi Modularity - Tutorial Retrieved 24/09/2012, 2012, from http://www.vogella.com/articles/OSGi/article.html

Chapter 7: Planning for the Impact of Distributed Solar Energy on the Grid

In this chapter, written as a conference paper[5], the implementation of two types of agents is described: the electricity consumption of individual consumers using historical load data for the different customer types (residential, commercial and industrial) and the electric output of a rooftop photovoltaic system subject to weather variability with the inclusion of cloud data from the Bureau of Meteorology (BOM). Simulations are presented that illustrate the type of analysis that can be performed with MODAM, discussing the results aggregated at the transformer level.

[5] The article was presented at the 2012 Solar Conference in Melbourne in December 2012. It received the Best post-graduate paper award

Planning for the Impact of Distributed Solar Energy on the Grid

Fanny Boulaire, Mark Utting, Robin Drogemuller, Anula Abeygunawardana, Gerard Ledwich, John Bell
Queensland University of Technology, 2 George Street, Brisbane, Queensland 4000, Australia

Planning for the Impact of Distributed Solar Energy on the Grid

ABSTRACT

The behaviour of single installations of solar energy systems is well understood; however, what happens at an aggregated location, such as a distribution substation, when output of groups of installations cumulate is not so well understood. This paper considers groups of installations attached to distributions substations on which the load is primarily commercial and industrial. Agent-based modelling has been used to model the physical electrical distribution system and the behaviour of equipment outputs towards the consumer end of the network. The paper reports the approach used to simulate both the electricity consumption of groups of consumers and the output of solar systems subject to weather variability with the inclusion of cloud data from the Bureau of Meteorology (BOM). The data sets currently used are for Townsville, North Queensland. The initial characteristics that indicate whether solar installations are cost effective from an electricity distribution perspective are discussed.

Keywords: Grid Integration, Agent-based Modelling, Solar City.

7.1 INTRODUCTION

The work presented in this paper is part of a large project which is using agent-based modelling to assist in understanding the impact of renewable energy sources on the electricity distribution system controlled by Ergon Energy in Queensland. This distribution system covers approximately 1.5 million square kilometres outside of the south east corner of Queensland. The focus of this project is on the physical infrastructure of the distribution network, rather than economic or market models.

This paper concentrates on solar energy issues. Currently, the major consideration for energy generators and distributors is to reduce peak load, as the current regulatory regime requires that the distributors achieve high levels of reliability on their networks. The current mix of electricity generation, with a high proportion of coal-fired capacity, also favours a stable base load with minimal variation. Obviously, if energy storage is not used, the prime consideration is in identifying those circumstances where solar energy can be harvested at times of high overall electricity demand.

The aim of the work presented in this paper is to understand the contribution solar panel output can have at different locations on a distribution network; performing analyses at a fine level of granularity, both spatially and over time. The contribution of this paper is the presentation of the methods used when analysing the benefit of increasing the level of solar panels to reduce peak consumption at a given location on a distribution grid. The analysis is performed using historical load data for the different customer types (residential, commercial and industrial) gathered at 30 minute intervals over a year (2010) and simulated for any node in the distribution network. Townsville was chosen as the first location for analysis because it was one of the locations chosen under the Solar Cities program, comparatively reliable data sets were available and Townsville has a high level of solar energy availability. Solar panel outputs are computed for that same time period and location, using the usual equations to estimate the solar energy output of the system (Duffie & Beckman, 2006); p11-15), as well as taking into account weather variability with the inclusion of cloud data from the Bureau of Meteorology (BOM).

Being able to simulate both consumption load and solar panel output for a standard 1kW PV system at a given point on the network for every ½ hour supports the identification of areas that would most benefit from the installation of PV. From these 2 simulated outputs, a number of analyses can be performed. Firstly, by comparing simulated load and simulated solar output, it is possible to understand the potential real contribution of solar panels at a given point in the network by calculating how much PV would need to be added to reduce the peak by given amounts. Secondly, having these simulations at fine level of detail, both in time and geography, accounting for the variation in the consumption from users as well as the production of electricity by PV systems, subject to actual weather information (cloud passing over the panels) can be useful especially since it can highlight those times that see extremes happen, which cannot be predicted when using the usual averages over large areas.

This paper introduces the simulation framework, describing the types of analyses possible, technically enabled through its modular structure. The methods specific to the analysis presented in this paper are then introduced. They describe the algorithms used to simulate the consumption load at any location in the network for any time of the year, as well as the expected output for a given 1kW solar panel under real weather conditions at a given location. Finally, results of an analysis for a selected transformer are given and discussed, showing the benefit of using this type of analysis when looking at shaving peak electricity demand for a given area of concern, from the point of view of a distribution company.

7.2 THE FRAMEWORK

A simulation framework is being built to assess the impact of the large-scale introduction of renewables on the physical behaviour of a distribution network. A range of analysis types are required in order to tackle such a complex question. Some of these analyses can be performed in isolation, while others need to be integrated in a holistic manner. In both cases, the technical characteristics of the electricity distribution grid are taken into account as well as the economic and sustainability challenges of minimising cost and carbon intensity. Consequently, this framework has been built using a modular approach, allowing the different types of analyses to be performed according to current needs. Representing the different elements of an analysis, including relationships to other analyses, allows the assessment of the impact of a change in one part of a system on other parts, which is a more accurate representation of what happens in reality.

The analysis presented in this paper falls into the category of analyses that can be performed independently, where the question being answered is: "***What is the capacity of solar generation that would reduce peak demand at particular points in the network?".*** With the platform built in a modular way, the simulation model to answer this question can be reused when performing more complex analyses. In the future, solar panels will be considered with other forms of renewable technology and extended to the interaction with battery storage.

A more complex analysis type, combining different analyses modules, is used when planning for a distribution network upgrade/maintenance, using simulated data at a fine level of granularity, both geographically and over time. Within this framework, an Agent-Based Model (ABM) and a Particle Swarm Optimisation (PSO) need to be integrated. ABM is used to represent the different system units accurately and dynamically, following the changes over time and at different levels of detail in the distribution network. The PSO module allows finding the most economical mix of network extension and integration of distributed generation over long periods of time (Boulaire, Utting, Drogemuller, Ledwich, et al., 2012).

The method and results of the analysis presented in this paper are part of this overall framework, implemented in a distinct module that can either be run independently or combined as part of larger, more complex analyses. The existing ABM part of the framework is used in combination with a newly implemented module that simulates the PV output at a given location under local weather constraints. The methods used for these two simulation modules are described below.

7.3 METHODS

Since measured time interval data is not available for most customers, simulation of load consumption for individual customers and output of solar panels at the customer location were performed. The area of study was limited to Townsville, Queensland; and the simulations were done for every half hour over a one year period (2010).

7.3.1 Simulation of the electricity load

As mentioned above, the framework contains a module implemented using an Agent-Based Modelling approach. In this module, the agents represent the different assets the distribution network whose behaviours influence the load consumed or flowing through the electricity distribution network.

This model was built using the following types of data:

- Network connectivity data: this is the information about the different assets (transformers, switches, lines, etc. at the Medium Voltage level) and the customers (Residential, Commercial or Industrial) that make up the network. Their characteristics and configuration are stored within the module. The topological relationships defining the structure of the network are also stored.
- Electricity demand data: three datasets were available and used for different purposes:
    - Simulation calibration
        - Quarterly billing consumption data for every customer for year 2010
        - Half-hourly consumption data for a limited number of customers for year 2010
    - Simulation validation
        - Half-hourly consumption data at feeder level for year 2010, for some transformers

These 2 types of datasets were combined so that for each asset described in the distribution network, the electricity consumed by it (in the case of customers) or flowing through it (for feeders, etc.) was simulated for each ½ hour over a given period of time. The setup of the simulation consists of 2 steps:

1. Each customer, located at a leaf of the distribution network is allocated a consumption profile from the set of available measured demand profiles, according to the algorithm described in Algorithm 1. The profile is scaled to match the quarterly billing information;

2. Each node in the network has the electricity load summed up as the branches are being connected up to the top of the tree.

Algorithm 1 – Description of the allocation of demand profiles for each customer in the distribution network.

The allocation of the interval data profiles to a customer is done as follows:

1. The tariff and the annual demand are obtained for each customer.
2. From the complete list of consumption profiles,
    a. a subset is extracted that contains all the profiles that have the same category (R, C or I) and tariff as the given customer. If no match is found:
    b. a subset is extracted that contains all the profiles with the given tariff, regardless of the customer category (from the exploration of the interval data and billing datasets, there is always at least one profile that is of the corresponding tariff)
3. From this reduced set of profiles, a subset is further narrowed according to a check on the profile average and peak demand against the customer annual demand such that:
    a. the average demand of the profile * 3 >= to the demand of the customer and (the peak of the profile < than 10 * the average profile or the peak of the profile < 10)
4. From this subset, a random profile is chosen
5. If there is no profile available for either the substation or the category from all the available profiles, no profile is returned and an exception is thrown.
6. The selected profile is scaled, so that the sum of the ½ hourly data over the year corresponds exactly to the annual billing data of each customer, ensuring accuracy in the annual estimation of electricity consumption.

A simulation over a given distribution network containing 92,261 asset agents was performed and output for every half hour over 20 years.

7.3.2 Simulation of the solar panel output

Many software packages (e.g.(H2RES MODEL, 2009; HOMER Energy, 2012; National Renewable Energy Laboratory, 2012)) are available to estimate the PV output of a standard solar panel, given its characteristics, at a given location. However, the weather files used in these types of software are based on a TMY

(Typical Meteorological Year). This can be useful when planning to install a solar unit and wanting to size it and see its economics for an average year.

However, in the context of this analysis, a more detailed model is required, so that the demand and the PV output can be compared at each time throughout a specific year. While the expected average of a PV output over a given time duration can be estimated, this can vary greatly depending on the level of cloud cover. Not only is the PV output dependent on the weather conditions, but in many cases, the electricity consumption is also dependent on the weather conditions. Correlating these requires real rather than normalised data. This happens for example when the load is greatly dependent on the use of air-conditioners, as is the case in many office based commercial customers. Consequently, in order to calculate the benefit of installing PV to support the load on the network, it is important to simulate its output at the same location as the load and with the same weather conditions.

Algorithm 2 was developed to predict the output of a solar panel at a given location.

This model was built using the following data:

- Weather information: this information describes the cloud levels at the beginning of each 3 hour period (in Octas: 0=clear .. 8=total), as well as the global daily irradiance (ranging 0 .. 35 MJ/m2) for a given location. This was obtained from the Bureau of Meteorology for the station of interest (closest to the area of analysis).
- Solar panel information: this information was chosen for a set type of solar panel. It is an input to the program and can be changed as required. The information needed is the rating, geographical information (latitude and longitude), azimuth, tilt, and derate factor.

Because of the nature of the data available, the algorithm proceeds in three steps. The first step estimates the available beam and diffuse irradiance for every half hour, based on cloud information. The results are expressed as a ratio (0.0-1.0) relative to the standard test conditions for solar panels: irradiance level 1000 W/m2, spectrum AM 1.5, cell temperature 25°C, and solar spectral irradiance per ASTM E 892 (Endecon Engineering & Regional Economic Research Inc, 2001). The second step calculates the proportion of this irradiance that actually falls on the surface of the solar panels, based on the time of day, time of year, location, azimuth and tilt of the solar panels (Duffie & Beckman, 2006). The third step calculates the expected AC output, given the rating and derate values of the PV system.

Algorithm 2 - Description of the calculation of PV system output at a given location, under given weather conditions, taking into account the passage of clouds.

For each date, and each ½ hour time, t:

1. Cloud values in Octas (0..8) are converted into percentage of irradiance, which was derived
   - from 2010 BOM data, as shown in Figure 1.
   - Calculate the direct and diffuse ratios:

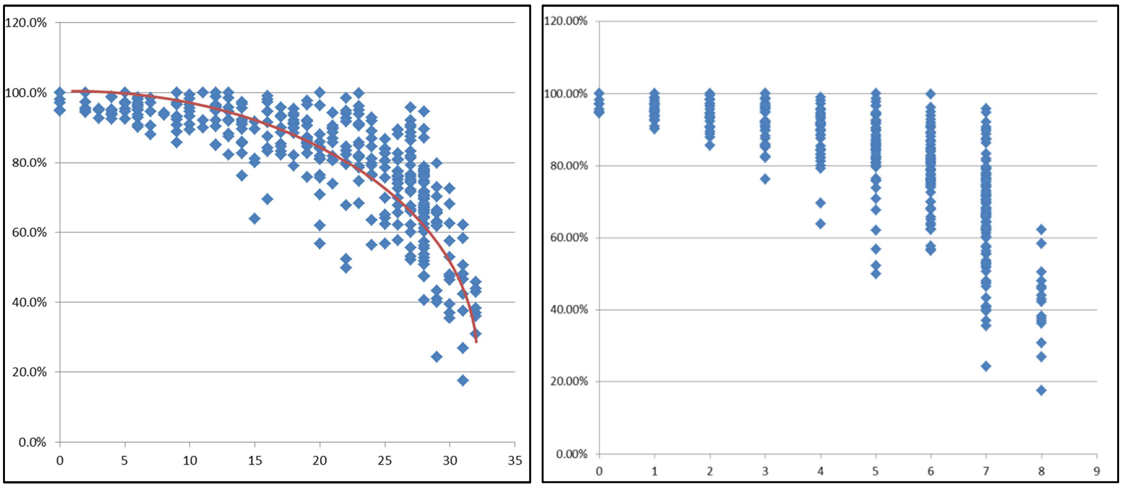
totalRatio = gaussian(mean(cloud(t)), stdev(cloud(t))) (see Table 7-1)

diffuseRatio = diffuseGraphLookup(totalRatio) (see (Liu & Jordan, 1960) Figure 5)

directRatio = totalRatio - diffuseRatio

1. irradiance(t) = irradiance(directRatio, diffuseRatio, date, t)
2. PV(t) = irradiance(t) * PVSize * DerateFactor

NB. the irradiance function takes into account latitude, longitude, tilt, azimuth etc.



(a) (b)

Figure 7-1 - (a) normalized global daily irradiance versus sum of daytime cloudiness values (6am 9am, 12pm 3pm) – (b) Same curve after clustering into 8 ranges

Table 7-1 - Mean and standard deviation of normalised global daily irradiance for each level of cloud coverage value

Cloud	0	1	2	3	4	5	6	7	8
Mean	96.78%	95.33%	93.93%	92.04%	88.42%	83.92%	79.10%	66.77%	41.55%
StdDev	0.0208	0.0238	0.0377	0.0565	0.0751	0.1065	0.1018	0.1421	0.1118

From Figure 7-1 (b), statistics on the percentage of global total irradiance were derived for each cloud level and are shown in Table 7-1.

After having determined the total global irradiance for each half-hour period based on random sampling of the cloud distribution, that total irradiance is then split into direct and diffuse components based on (Liu & Jordan, 1960).

Example outputs from steps 1 and 2 on the 1st of January 2010 (with three-hourly cloud data of 8, 1, 8, 8, 8, 7, 7, and 7) are given in Figure 7-2.

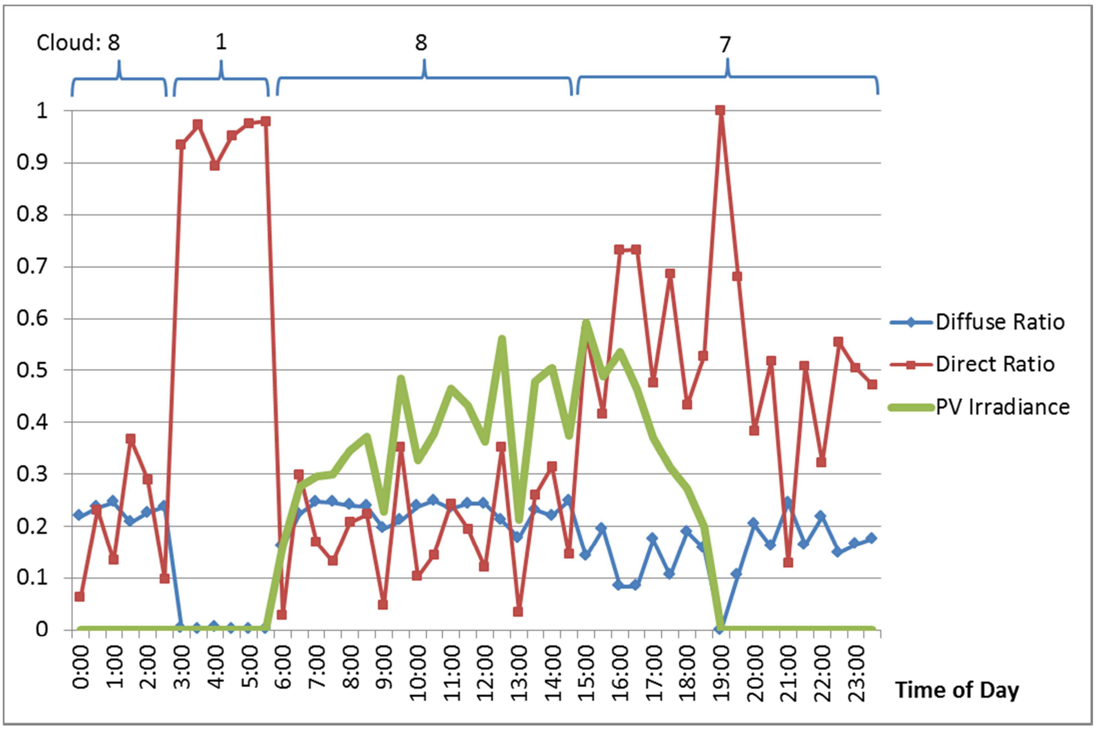


Figure 7-2 - Example output from steps 1 and 2 of the PV algorithm for 1 Jan 2010. Step 1 produces the disaggregated direct and diffuse irradiance ratios throughout the whole 24 hour period. Step 2 produces the irradiance falling on the PV panels. All results are ratios relative to standard test conditions.

Figure 7-3 shows some example outputs from step 3 with a PV rating of 2kW and a derate factor of 0.8 for the 1st of January 2010, with two different random seeds, and three levels of cloudiness (0, 7, 8).

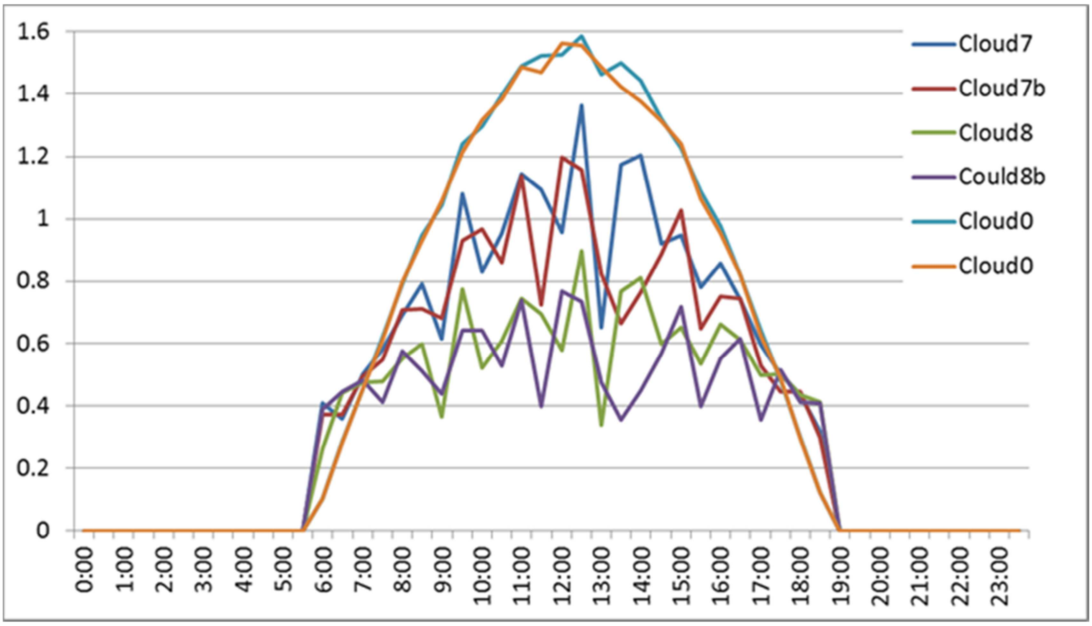


Figure 7-3 - Example outputs from step 3 of the solar simulation algorithm using 2 seeds and 3 cloud levels.

7.4 RESULTS AND DISCUSSIONS

Using the methods described above, 30 runs were performed for both the simulation of the load at a given transformer in Townsville, and the solar panel output at that same location for year 2010. The solar panel characteristics were: latitude (19.2500° S), longitude (146.8000° E), tilt (9degrees), azimuth (180 degrees, that is facing north), derate factor (0.77), rating (1kW); and all the calculations are done for a 1kW solar panel.

7.4.1 Identifying the number of solar panels required to reduce peak load

Using the simulation runs of both PV and load, the average over each of the 30 simulations was calculated for each ½ hour over 2010. These were then plotted against one another as shown in Figure 7-4.

If the load was exactly correlated with the PV output, all the points would lie on a straight line, meaning that at any time during any day of the year, the PV output would be exactly proportional to the load. Obviously this is not realistic. An interesting characteristic of these graphs however, is in their shape. The heavier the cloud of points towards the far right top corner of the graph, the more likely the peak consumption will be clipped. This can then be used to size the PV capacity for that particular location by dividing the load to be clipped by the proportion of the output of a 1kW PV. This gives the number of 1kW units that would need to be installed to provide that level of peak clipping.

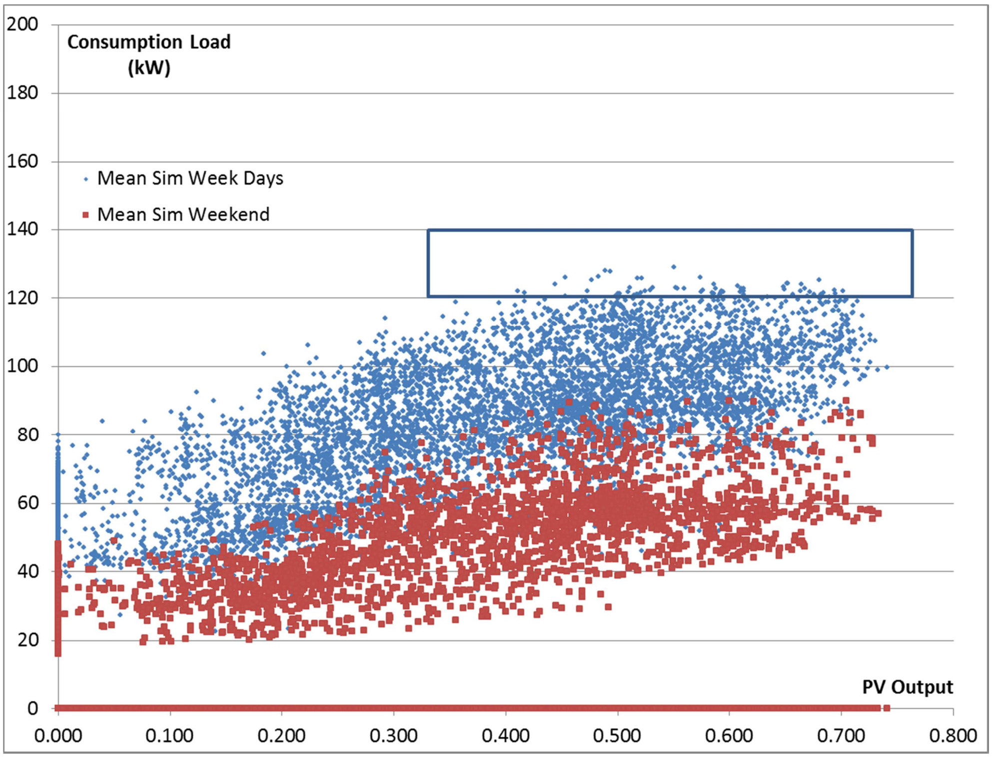


Figure 7-4 - Simulated consumption load versus simulated PV output.

Considering the example given in Figure 7-4, a rectangle has been drawn which corresponds to the points to be clipped. The aim is to reduce the peak load under 120 kW. The amount of PV capacity required will correspond to the following equation:

$$Max_t\left(\frac{P_t - ClippingValue}{PV_t}\right)$$

Where P_t is the peak at time t and $ClippingValue$ is the maximum desired load on this transformer (120kW in this example); PV_t is the PV output at time t for the 1kW solar panel described above under weather conditions at time t.

This will then give the maximum size of PV required, which will then be used to inform the economic analysis. Note that if demand is high (> $ClippingValue$) when PV is zero, the result says that an infinite amount of PV is required, which shows that PV cannot be used to reduce peak demand. This is typically the case for residential feeders, where the peak demands tends to be between 6pm and 8pm.

In this example, the maximum load for the simulation is 128.754 kW, requiring a clipping value of 8.754 kW. At the peak load time, the PV output was estimated at 0.550 kW. Consequently, a PV of at least 15.9 kW capacity is required.

Having this increased capacity of PV, the simulation can be run again in order to examine load reduction at other times of the year. While this number gives the maximum required PV capacity, intermediate levels may be used to provide a more balanced economic perspective.

Based on the amount of PV required to reduce the peak demand to various levels, an economic analysis of future upgrade options was performed using GAMS/CPLEX (GAMS Development Corporation, 2012), assuming an annual peak demand growth of 3.8% and a transformer capacity of 170kW (i.e. 200KVA with a power factor of 0.85). The demand will exceed the transformer capacity in 2019, and will eventually require a transformer upgrade, but for this transformer it is possible to defer that upgrade by installing PV. Assuming a PV capital cost of $5000/kW, the optimal economic solution is to install 8kW of PV in 2019, and defer the transformer upgrade until 2020. With a lower PV capital cost of $3500/kW, the optimal solution is to install 8kW of PV in 2019, an additional 9kW in 2020 and a further 9kW in 2021, thus deferring the transformer upgrade until 2022.

7.4.2 Validation of simulations

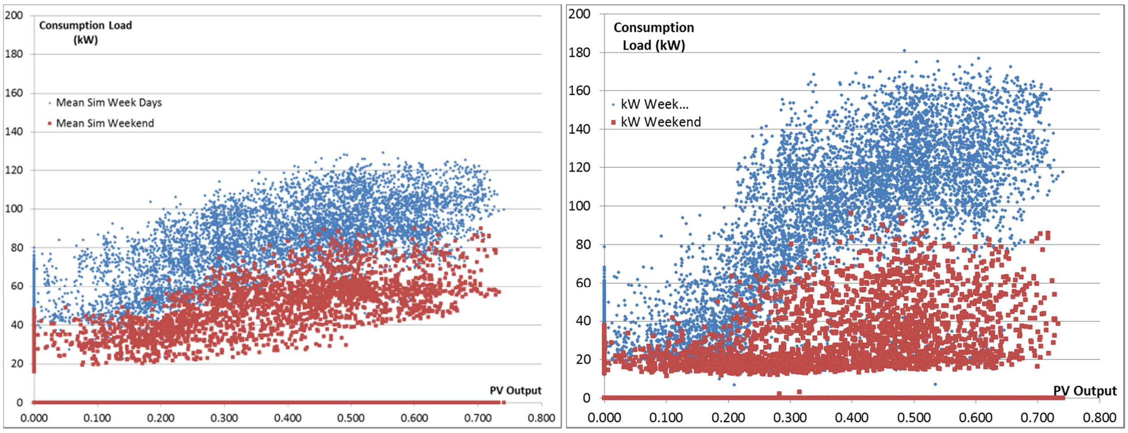
The simulation results were validated against measured data for the transformer used in this example. However, no measured solar panel output data was available for that location. Figure 7-5 shows two graphs of the load consumption versus simulated PV output, (a) being the simulated load and (b) the measured load. The data was further distinguished according to whether the load is consumed during the week or the weekend, indicating in a sense the type of business the transformer covers. As shown in this figure, the simulated data did not exactly match the actual load.

The percentage error was calculated for every data point (every ½ hour) and these errors averaged to 30% over the simulation year (2010), while the total yearly consumption error amounted to 3%. Such discrepancy in the results can be explained by the choice of the simulation algorithm. Indeed, as described in the Methods section, the simulation for the load consumption relies on the appropriate selection of existing profiles, and scaling them in order to obtain the accurate yearly consumption. While the overall yearly consumption is accurate, this approach can lead to varying results depending on the selected profiles. No information is available at this stage regarding the provenance of individual demand profiles, due to privacy issues, except from the knowledge that it is a customer of type Commercial. Consequently, significant differences in results can be obtained.

Figure 7-6 illustrates this type of difference in simulation output. Two simulations have been selected that give very different shapes in the results. In the first case, Figure 7-6 (a), the simulation seems to be dominated by two types of commercial customers, one that has a weekend pattern while the other one seems to have a continuous pattern of activity (some red dots have high values, similar to blue ones). In Figure 7-6 (b) however, there is very little difference between weekly and weekend patterns suggesting that profile of commercial customers with continuous load consumption have been selected.

In order to reduce these types of error, more information about the commercial customers is required. As such, it would be extremely useful when dealing with commercial customers to have information at least relating to some characteristics of the business (Monday to Friday opening hours or 7/7, seasonal patterns or yearly constant, 9-5 or 24 hours…), at least narrowing down the profiles to be allocated at the right location.

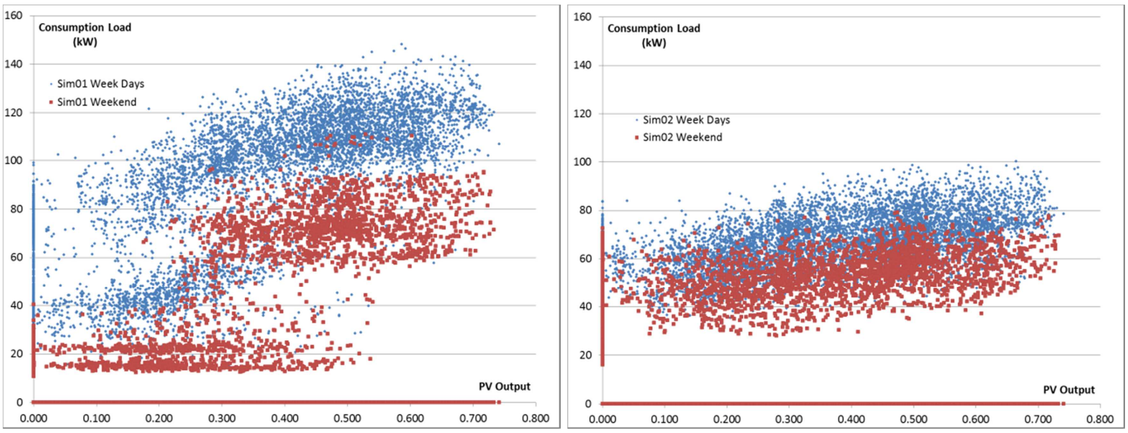
Requests have been made to the holders of the input data for higher levels of customer characterisation. This should be available shortly.



(a) Simulated Consumption versus Simulated PV output

(b) Measured Consumption versus Simulated PV output

Figure 7-5 - Simulated and Measured Load Consumption versus simulated PV output.



(a) Influence of 2 types of Commercial Loads (weekend pattern and continuous pattern)

(b) Influence of a Commercial load that is constant regardless of weekends

Figure 7-6 - Example of 2 simulated consumption outputs versus simulated PV output.

7.5 CONCLUSION

This paper has demonstrated the successful use of methods to analyse the benefits of introducing solar panels into groups of customers, or directly into the electricity distribution network. While the data used in the analyses was for

Townsville, Queensland, the method used has wider applicability. The results presented here were for a given transformer, but this analysis can also be done at the customer, feeder, distribution substation and zone-substation level. Also, different orientations of PVs have been tried and only the results for one are presented here; comparison of best PV orientations can also be performed.

The next steps will be to extend the analyses to larger groups of customers and transformers/sub-stations. Exhaustive search is unlikely to be appropriate once the population size scales up. Fortunately another module within the overall software framework identifies power system components that are overloaded. This provides the potential for heuristic driven search of the solution space.

Another area of future work will be in increasing the accuracy of the simulation output for consumption load. As discussed in the results section, the simulation output for the load consumption can vary depending on the customer types allocated within the simulation. This could be improved if additional information regarding the customers were available, such as information relating to the activity type of pattern (Monday to Friday opening hours or 7/7, seasonal patterns or yearly constant, 9-5 or 24 hours). Having such information, allocation of profiles could be done more accurately using clustering of the available profiles according to these characteristics. Other methods for load estimation at the customer level could also be implemented, such as regression analysis on factors that are influencing consumption (e.g. day of the week, weather conditions). Such methods have already been applied in the context of the larger project, but for residential customers only. These will now be applied for those commercial customers whose peak load is mainly driven by air-conditioners, such as offices.

7.6 ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the State of Queensland acting through the Department of Employment, Economic Development and Innovation (Employment Industry Development and Innovation) and the National and International Research Alliances Program for their financial contribution, the contributions of diverse partners on this project and especially Ergon Energy for providing the data used in this analysis.

7.7 REFERENCES

Boulaire, F., Utting, M., Drogemuller, R., Ledwich, G., & Ziari, I. (2012, 9-12 December 2012). A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network. Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.

Duffie, J. A., & Beckman, W. A. (2006). Solar engineering of thermal processes. Hoboken, N.J: Wiley.

Endecon Engineering, & Regional Economic Research Inc. (2001). A Guide to Photovoltaic (PV) System Design and Installation (pp. 1-39): California Energy Commission.

GAMS Development Corporation. (2012). GAMS Retrieved 05/10/2012, 2012, from http://www.gams.com/

H2RES MODEL. (2009). H2RES Model Retrieved 03/05/2012, 2012, from http://powerlab.fsb.hr/h2res/

HOMER Energy. (2012). Energy Modeling Software for Hybrid Renewable Energy Systems Retrieved 03/05/2012, 2012, from http://www.homerenergy.com/

Liu, B. Y. H., & Jordan, R. C. (1960). The interrelationship and characteristic distribution of direct, diffuse and total solar radiation. Solar Energy, 4(3), 1-19. doi: 10.1016/0038-092x(60)90062-1

National Renewable Energy Laboratory. (2012). PVWatts - A Performance Calculator for Grid-Connected PV Systems Retrieved 05/01/2012, 2012, from http://rredc.nrel.gov/solar/calculators/PVWATTS/version1/version1_index.html

Chapter 8: Impact of Technology Uptake on an Australian Electricity Distribution Network

This chapter has been written as a journal paper[6]. It illustrates how MODAM can be used to run complex simulations by mixing and matching agents to assess the impact of technological change on the distribution grid. The rooftop solar panel agents, described in the previous chapter, are used in conjunction with electric vehicle agents that have been added to the model, demonstrating how the model can be extended. Their implementation is described where two algorithms are available, showing the flexibility in the model definition. Finally, a scenario is used as input to the simulation, from which four cases have been derived, to illustrate the flexibility in building simulations. It also shows the potential these simulations offer to understand how a technology might impact a network depending on the way it is used.

DOI: 10.1016/j.envsoft.2015.03.019

[6] This paper was published in the Journal of Environmental Modelling & Software

Impact of Technology Uptake on an Australian Electricity Distribution Network

Fanny Boulaire, Mark Utting, Robin Drogemuller
Queensland University of Technology, 2 George Street, Brisbane, Queensland 4000, Australia

Impact of Technology Uptake on an Australian Electricity Distribution Network

ABSTRACT

This paper presents simulation results for future electricity grids using an agent-based model developed with MODAM (MODular Agent-based Model). MODAM is introduced and its use demonstrated through four simulations based on a scenario that expects a rise of on-site renewable generators and electric vehicles (EV) usage. The simulations were run over many years, for two areas in Townsville, Australia, capturing variability in space of the technology uptake, and for two charging methods for EV, capturing people's behaviours and their impact on the time of the peak load. Impact analyses of these technologies were performed over the areas, down to the distribution transformer level, where greater variability of their contribution to the assets peak load was observed. The MODAM models can be used for different purposes such as impact of renewables on grid sizing, or on greenhouse gas emissions. The insights gained from using MODAM for technology assessment are discussed.

Keywords: Agent-based modelling, electricity demand, distribution network, decentralised generation.

8.1 SOFTWARE

MODAM (MODular Agent Model) is a simulation environment that was developed to support building large-scale ABMs using a modular approach, with the aim of assessing the impact of different trajectories of consumption at different locations of the electricity distribution grid over many years. It has been developed as part of a research project led by Queensland University of Technology (QUT) since 2011, which was funded through a NIRAP (National and International Research Alliance Program) grant from the Queensland Government. This software is currently under review by the project partners to become open-source.

This software has been developed using Eclipse technology, and is written in Java. It has been exported to be used on Windows and Linux environments. Table 8-1 shows statistics on the software tools.

Table 8-1 - Development statistics for the NIRAP software tools

	Program		Automated Tests	
	Number of Classes/Files	Source Lines of Code (SLOC)	Number of Classes/Files	Source Lines of Code (SLOC)
MODAM ABM Framework (Java)	84	13,160	46	4,650
Electricity Network Assets and Agents (Java)	196	37,700	119	16,100
Visualisation Generation Programs (Java)	5	1390	1	151

The electricity grid model developed with MODAM supports more than 7000 combinations of parameters and agents descriptions. We used the pairwise test design strategy to check that each pair of compatible modules and options were tested together.

Two developers worked on this software:

- Fanny Boulaire
    - QUT, P Block, 2 George Street, Brisbane, Qld 4000, Australia
    - +61 7 3138 4508

    - Fanny.Boulaire@qut.edu.au
- Mark Utting
    - QUT, P Block, 2 George Street, Brisbane, Qld 4000, Australia
    - +61 7 3138 0773
    - Mark.Utting@qut.edu.au

The data used in the simulations presented in this paper is not publicly available, as it is based on network data owned by Ergon Energy, which is the industry partner on this project. Only the exogenous scenarios, developed using R can be accessed.

8.2 INTRODUCTION

Australia has seen a steady increase in electricity consumption over the last few decades, with an average of 2.8% a year over 2000-2010 in Australia (Cuevas-Cubria et al., 2010). Trends had been set to carry on increasing; however, they have reversed since 2010 and it seems they will continue decreasing or stabilise, at least in the near future. Could the measures taken by the government to reduce pollution and carbon emission have started to make an impact on the electricity consumption patterns? Measures have been introduced to encourage take-up of renewable technologies (Australian Government - Clean Energy Regulator, 2012b; Australian Government, 2011b) and have been well received by the population, which has resulted in an increase in the number of small-scale generators (Queensland Government - Office of Clean Energy, 2011), mainly rooftop photovoltaic (PV). However, according to (Saddler, 2013), the main drivers for this decline are changes in the economy away from electricity intensive industries (shutdown of some industries), and the response of electricity consumers to higher electricity prices (since 2010), which are more a reflection of the economy state than the result of active policies. However, the impact of energy efficiency programs (efficient appliances, solar water heaters…) was also mentioned as a large contributor to the reduction, and while not considered as the main driver of the reduction in national demand, decentralised generators might still have contributed to the reduction in energy required to be produced by the centralised system. Because of their decentralised nature and the fact that their production is consumed locally, their contribution is hard to quantify and is not accounted for at the national level. However as their number is still expected to increase, these technologies might become a bigger driver in the change of electricity consumption and the move away from coal-generated electricity to renewable sources. In addition to renewable energy, storage technology has been identified by (Manyika et al., 2013) as one of the 12 disruptive technologies that could have a big impact on the economic and societal landscape by 2025, leading to further changes to the electricity sector in the near future. While labelled disruptive, these technologies can however have a positive outcome on the environment and the electricity grid, if their characteristics are properly understood and their potential harnessed.

Understanding how the introduction of decentralised technologies might impact their network is one of the aims Ergon Energy, one of the two electricity distribution companies in Queensland, had when commissioning us with a project. With an energy infrastructure transitioning from a centralised to a decentralised system, new patterns of consumption are starting to emerge. Customers who used to only consume are now producing thanks to the introduction of rooftop photovoltaic, and they might further modify the traditional consumption patterns with the use of batteries and electric vehicles which are starting to gain popularity. Understanding where, when and how peak consumption might be changing is important to ensure the distribution network is sized appropriately, so that electricity is distributed in a sustainable, safe and economic manner. Because of these new technologies, the past is no longer a good predictor of the future, and new methods for predicting electricity consumption are required.

We have been developing a simulation environment to assess the impact of different trajectories of consumption at different locations of the electricity distribution grid over many years. This simulation environment, called MODAM (Boulaire et al., 2013a; Boulaire, Utting, & Drogemuller, 2015a), is built using agent-based modelling where the different elements of the distribution network are represented in terms of their physical characteristics as well as the way they can be used, through explicit description. As new technologies (e.g. solar panels, batteries), new policies (e.g. time-of-use tariffs) or/and demand management are being introduced in the network, understanding how these changes will affect the assets at every node in the network can be done by developing scenarios of possible future using a whole-of-system approach. Through such an approach, the repercussion of one change on the rest of the network can be accounted for, as well as coincidental changes; this might give a very different picture of the future of the network than if considered separately.

This paper introduces MODAM briefly, describing first the platform and the way it was built, using a modular approach. The different elements making up the system representation are then described, and how to build a simulation is also explained. An application of our simulation platform is then given, which is the main focus of this paper. For this, four simulations were performed based on scenario B developed by the Future Grid Forum (FGF) (CSIRO Future Grid Forum, 2013). This scenario, called "rise of the prosumer" expects to see a rise of on-site generation of electricity, using rooftop photovoltaic (reaching a capacity of 46% by 2050), and an increase in the number of electric vehicles (reaching 27% of the vehicle fleet by 2050). This scenario was used as an exogenous scenario to our agent-based model, whose development is explained. Information about the simulation runs is given followed by an impact assessment of these scenarios. Finally, the use of agent-based modelling for technology assessment is discussed with particular reference to the insights gained from the MODAM model.

8.3 RELATED WORK

A vast number of forecast models of electricity demand have been developed using different methods. In (Shukla, 2013) the author compares two approaches that have been used to model the energy sector: top-down (macro-economic) and bottom-up (techno-economic) models. Some top-down models are based on computable general equilibrium theory, assuming perfect market equilibrium conditions, and often only representing monetary flows and no physical flows of energy or other commodities. Other types of top-down models are statistical models that forecast demand also based on economic variables but also other variables of interest. An example of this is the model of peak half-hourly electricity demand forecasts developed by (Fan & Hyndman, 2013) for Queensland. Using data from AEMO (Australian Energy Market Operator) over more than 10 years, the authors have developed a model that includes temperatures, calendar effects, demographic and economic variables. These forecast distributions are done using statistical methods based on understanding the past to predict the future over the whole of Queensland. While geographical variability is considered in terms of temperatures at three locations, no other local information is used.

Bottom-up models that are based on techno-economic perspectives and often reflect technological progress follow two kinds of computational frameworks: optimisation and accounting (Shukla, 2013). Optimisation models such as MARKAL have been used for the purpose of energy technology research and development planning (Rath-Nagel & Stocks, 1982). Accounting models, which are driven by exogenous assumptions about the energy demand and supply, are often carried out to quickly assess the impact of policy options, as well as for back-casting, to find a pathway to achieve a set of future goals (Shukla, 2013).

While technological change in infrastructure has traditionally been modelled using optimisation and equilibrium models, agent-based modelling, which is a bottom-up approach, has recently gained some popularity for its capacity to see "what could be" under different scenarios rather than "what should be" which is what optimisation brings (Ma & Nakamori, 2009; Veneman, Oey, Kortmann, Brazier, & De Vries, 2011). Agent-based modelling is well suited to capture the actions and interactions of the different elements, or agents, that form a complex system. Different traditional approaches are commonly used for modelling complex systems which include but are not limited to systems dynamics, discrete-event simulations, participatory simulation, Bayesian networks, knowledge-based models, statistical modelling and risk analysis (Kelly et al., 2013; North & Macal, 2007). These techniques have different properties that are most suited to some applications; choosing which modelling technique to use can be guided by identifying the purpose of the modelling exercise (e.g. forecasting, system understanding, social learning), the type of data (qualitative or quantitative), and how space, time and entities (or structure) of the systems are conceptualised (Kelly et al., 2013). Using these criteria, agent-based modelling was identified as the choice modelling technique for our research for its capacity to capture information at a fine level of detail over space and time using simple rules in the aim of understanding how the system responds to changes from the environment and the entities' responses and interactions. Initially described as swarms in (Bonabeau et al., 1999), this type of response of the system is described as emergence which is a key feature of ABMs. Such phenomenon can also be observed in the context of the electricity distribution. Indeed, the interactions of the different actors such as consumers, solar panels, and batteries at the premise level can influence the flow of electricity at the zone substation depending on their consumption, the environmental conditions and the battery control algorithms chosen which adapt to stimuli from the system. This characteristic of emergence is one of the key aspects to justify the use of ABM in our research. By describing what happens at the micro level and see what happens at a macro level we might be able to avoid "surprises" (Hall, 2011) by being more aware of the trajectory the system might take when people are adding more solar panels in some specific areas or when new technologies will start to really have an impact (e.g. with the introduction of electrical vehicles or small-scale batteries).

Agent-based models have been mainly used to understand electricity markets when deregulation happened and to help better design them (Batten & Grozev, 2006; Conzelmann et al., 2005; Foley et al., 2010; Nikolic & Dijkema, 2010; North et al., 2002; Weidlich, 2008; Zhang, Zhang, & Bi, 2011), however its use has lately broaden, with applications investigating the impact different technologies and policies (Cai et al., 2011; Chappin & Dijkema, 2010) or demand-side management measures (Boait et al., 2013) might have on the system as a whole. In the context of this research, understanding the impact of technologies on the distribution network where the physical infrastructure is represented is important. Few studies have been published that model both the infrastructure of the distribution network and the actors impacting it using agent-based modelling (Cai et al., 2011; Institute for Energy and Transport, 2014), however in the case of (Institute for Energy and Transport, 2014) it is unclear at this stage how the agents and the infrastructure are modelled. The work presented in this paper belongs to this class of research, which it extends by assessing the impact of a larger range of technologies that can impact the functioning of the distribution grid.

Rooftop photovoltaic (PV) and electric vehicles (EVs) are two technologies of interest when it comes to assessing the technologies that are impacting our distribution grid (e.g. PV), or have the potential to (e.g. EV). These two technologies can be used together in order to support the network, as demonstrated in (Alam, Muttaqi, & Sutanto, 2013) where storage devices, using adapted charging/discharging strategy, in partnership with PV can mitigate the impact of voltage rises when high levels of PV are installed as well as to support the evening peak load. As PV has been in place for some time now, it has been quite extensively studied (Pezeshki, Wolfs, & Johnson, 2011; Tonkoski, Turcotte, & El-Fouly); however, EV is a more recent technology and less is known. Nonetheless, a few studies on the potential impact of EV in Australia and the necessary changes that will be required on the power networks exist (Dow, Marshall, Le, Aguero, & Willis, 2010; Higgins, Paevere, Gardner, & Quezada, 2012; Paevere et al., 2014; Ustun, Zayegh, & Ozansoy, 2013). These show that EV will have a significant impact on the distribution systems that the distribution companies need to take into account. Indeed, the addition of two EVs on the distribution network would correspond to the addition of one new house to the neighbourhood (Dow et al., 2010), and because

85% of Australians own a car with 60% of household having 2 or more cars, this means that a full migration to EV would correspond to a 30% rise in the number of houses in a neighbourhood. This is not negligable, especially since these EVs might influence the peak household electrical load in a greater manner, depending on the charging modes (Paevere et al., 2014), which in turn will impact the distribution grid (Ustun et al., 2013). Knowing where these EV might be taken up can be informed from uptake of EV studies over highly granular geographical areas (Higgins et al., 2012) which provide electricity providers with a better insight into where to focus their efforts. However, these results do not link to the infrastructure, which is important for the planners to size the network appropriately, as already noted back in 1999 in a report for the European Commission that calls on expanding models beyond energy-environment-economy models to include network structure and agent behaviour (Grohnheit, 1999). Some work has been undertaken to understand the impact of EV using controlled and uncontrolled charging at the feeder level (Dow et al., 2010), showing that the distribution system started having problems at penetration levels of EV below 5% for the uncontrolled scenario. Following their results, the authors concluded that the impact of plugin electric vehicles (PEVs) on the different areas of service should be understood by the utilities, and that PEV adoption rates could be predicted to support planning of required additions, on a local-area basis.

From these different studies, it is higlighted that there is a need to model the impact of technologies at a fine level of detail in terms of space and time, and that these models need to have their demand linked to the network infrastruture. The work presented in this paper fills this gap, where an analysis is done at a local level and the infrastructure is taken into account. The impact of two technologies (EV and PV) is assessed using agent-based modelling, which is a technique that has proven to be useful when modelling technological change in infrastructure.

8.4 OVERVIEW OF THE AGENT-BASED MODEL FRAMEWORK

MODAM, a simulation environment, has been developed to support agent-based modelling of the impact of different trajectories of consumption at different locations of the electricity distribution grid. This tool was built with the requirements of flexibility and extensibility so that a large number of models and simulations could be created during the research project, but also to permit existing models to be used, and new models to be created by others, beyond the timeframe of the project. Before deciding on the implementation of MODAM, two existing software systems, Repast (Argonne National Laboratory, 2014) and MASON (Luke et al., 2005) were used to implement our modelling and simulation application. They however showed limitations in regards to some of our requirements, especially in the way the simulations are setup. Both systems have a central location for the setup of their simulations where a modeller needs access to the code, when one of our aims was for a planner to run simulations on a daily basis without the need to program or access code. Further, our need for quick assessment of the impact of different technology and behaviour options on the network meant that many options needed to be handled not only for the data but also the sub models, which can quickly become complex when having all these options described in a central location. Additional details regarding our choice in developing MODAM and how this is done can be found in (Boulaire et al., 2015a). A concise description of the system is given here.

The simulations performed with MODAM are discrete time simulations, using synchronous (time-stepped) time advance mechanisms where a model proceeds in half-hourly timesteps. The agents described in the models are heterogeneous agents, as they perform different roles and are subject to different objective functions, which are represented by simple rules. Individuals know their own characteristics that are influencing their own output values, as well as the environment they are in and the other agents they are interacting with. Agents are also influenced back by the environment they are in, and the agents they interact with. Interactions are modelled explicitly at the individual level and each entity knows what environment it is in, and the other entities it is connected to, through an underlying network structure representing the distribution network.

As the model will evolve through the actions and interactions of the different agents, that is, in an endogenous manner, exogenous scenarios can also be implemented to influence the system. These can be developed as models to be input to the simulation, or they can simply be parameters to the model.

MODAM's architecture and the models built using it were developed following a modular approach. This means that the software framework was developed following practices from component-based software engineering (Szyperski, 1997), where the different functionalities of the modelling and simulation framework were separated (e.g. separation of the simulator, the model, the user interface and the databases). Further, the development of the model representing the electricity network under study was built so that the different agents created in distinct components are weaved together to form a homogeneous model at simulation setup only. This is done in an automated manner within MODAM, and a modeller needs only calling the different components of interest for the study, which contain the description of the agents' characteristics and rules without having to code; see (Boulaire et al., 2013a, 2015a), for details on the compositional implementation of MODAM. When extending the model, a programmer will implement new agents in independent components, or plugins, following the system's processes so that these can be weaved together in an automated manner at runtime. These agents can also have some of their information populated through data providers that can also be developed in separate plugins. This approach was taken so that a non-programmer could develop different models and simulations with different types of agents that not only have different input values but also different structures (agents placed differently over the networked structure, as well as with different behaviour logics). Further, it was developed so that the model follows the principle of "*start simply, verify, validate and grow the model*" as mentioned in (Banks & Chwif, 2011), where a base model can be built, verified and validated over smaller time intervals during the project life, continually extending as information becomes available while providing confidence on the way it is developed. This approach has the additional benefit that the model can continue on expanding beyond the timeframe of the project to make management decisions when new technologies are becoming adopted by consumers. In this case, a programmer would need to implement the new technologies as agents, but they would not need to go into the previously written code and would simply add the class of agents to the model, and if required, the data providers to populate them.

Below, in Table 8-2, are described the different elements currently implemented in the network model within MODAM. Two entities can be distinguished here that compose an agent: the assets and the behaviours. The assets are the different physical entities that form the underlying network structure. The behaviours contain the sub models and private data on which they make their decisions. The behaviours are not bound by structure directly but access the underlying network through their asset. This separation of assets and behaviours resulted from the requirements of extensibility and flexibility, which allowed breaking down further the independence of the implementation of behaviour and data population. Each of these assets and behaviours can be defined within distinct plugins. Depending on the aim of a simulation, only the plugins of interest will be called upon at simulation setup, and within each of them only specific classes of behaviours, for example, may be selected. This is the case for example in this paper, where not all assets and behaviours described in Table 8-2 will be used. Battery agents (battery assets with their associated battery strategy and control behaviours) are not part of the simulation, for example; however, they are still available in the definition of the model and can be called upon for another type of simulation. The simulations can be populated using different datasets, which can inform the number of agents and their characteristics that will be part of a model on which to run simulations, as well as the network on which they will be performed. In addition to the agents' actions and reactions, these can also be subject to the environment, described in Table 8-3. The weather information is used by a few agents to inform their behaviour, such as rooftop solar PV or premises whose consumption will vary depending on the use of technologies with seasonal influence (e.g. air-conditioners, or heaters). Finally, different types of networks can be modelled: medium or low voltage, as well as Urban (3 phase) or SWER (Single Wired Earth Return) networks.

Assets	**Behaviours with sub models**
Network Assets: • Bus; • Transformer; • Switch; • Line	Load model: • Load summation model; • Global voltage analysis algorithm (load flow)

Premise Assets	Load model: • Based on historical data from individual premises; • Weather-driven model; • Historical data from feeders; • Combination
Battery Assets: • Grid Battery; • Premise Battery	Battery Control • Fixed Time of charging and discharging; • Variable times learnt from previous data; • Communication to feeder; • Independent from network
	Battery Strategy • Constant Discharge; • Load Following
Solar Assets: • Rooftop Solar PV Assets	PV output model • Weather informed; • Based on historical data
Electric Vehicles Assets • Battery EV, Plugin Hybrid EV	Charging method • Controlled charging; • Uncontrolled Charging

Table 8-2 - Assets and behaviours, forming the agents currently implemented in MODAM. An asset can have one or many behaviours to describe its rules that can be used in combination or independently.

Environment	
Weather information	• Historical data from Bureau of Meteorology; • Typical Meteorological Year (TMY)
Network Type	• Low voltage, Medium voltage; • Urban network, SWER (Single Wired Earth Return) network

Table 8-3 - Environment data the agents evolve in.

Thanks to the separation into different plugins, the agent-based model can be extended quite easily when new technologies or policies need to be added; and many combinations of behaviours can also be trialled and compared to see the impact they might have on the system overall. At this stage of our implementation, more than 7,000 combinations of behaviours are possible, which is rather a large number to test. To ensure that we systematically tested these combinations, we used the pairwise test design strategy (see http://www.pairwise.org - we used the 'Jenny' tool) to check that each pair of compatible modules and options were tested together. In addition, unit tests are performed on every class created in their own plugin for model verification, ensuring the correctness of the implemented operations that each agent is subject to. Validation of the model is also performed using quantitative methods (Bennett et al., 2013) where simulation outputs are compared to recorded data for different metrics. This is done at the unit level for modules that can be used individually (e.g. for the solar panel module), or/and by combining some modules to represent a given system (e.g. for a low-voltage network, such as a street, for which household consumption and solar panels are modelled). In both cases, simulated outputs were compared to actual records, which are time-series of the state variable of interest. The assessment was done at the agent level, or at another point within the network which was the result of the aggregation of a pool of agents (e.g. at the transformer or feeder level). Quantitative methods such as direct value comparison were performed (scatter plots comparisons, or other metrics such as mean, range, variance) as well as qualitative methods where expert opinion was obtained from the project partners. This qualitative assessment had the additional benefit of enhancing communication with the partners and refining the requirements of the model.

Depending on the simulation, the number of agents varies, but the framework is set as a large-scale ABM where thousands of agents can co-exist. As the simulation runs, population dynamics emerge, which are observable thanks to the definition of different types of state variables. Examples of these are the load, the real and reactive voltages, and the real and reactive currents, which inform the state of the network at each node in the network over a given period as chosen by the user. Stochasticity in the simulation is obtained through the allocation of load profiles for individuals, as well as location of some new technologies such as PVs, electric vehicles, batteries... Load profiles from actual records were used as input to the model and allocated randomly to the consumers located at the leaves of the network. The random allocation was however guided according to some criteria so that it was as close to reality as possible. The location and type of technologies can also be done in a random manner, through the guidance of socio-economic criteria over the areas.

The state variables can be observed as the simulation is running, via inbuilt facilities to graph any variable of interest, and they can further be used for different types of assessments post-simulation, as well as input to other types of models, because they can be saved in csv files. As an example, load data at each transformer over the simulation period is used as input to an optimisation program that is run as part of this project, to find out the most economical upgrade and extension of the distribution network.

As can be seen here, many types of simulations can be performed. The following section describes in detail how four simulations were built using this framework, and details of the results are given.

8.5 USING MODAM TO MODEL TECHNOLOGICAL CHANGE IN INFRASTRUCTURE

This section gives a detailed example of the use of MODAM. A scenario, developed by the Future Grid Forum (FGF) (CSIRO Future Grid Forum, 2013), was chosen as our input to the ABM, and further developed into four simulations in the aim of understanding how this possible future might impact the distribution network, depending on where and how technologies are used. The initial scenario is first described in this section along with how it was extended to describe the four simulations, run on areas of Ergon Energy network in Townsville. Then, a description of the system is given and the simulation set up requirements, followed by an explanation of how the scenario from the FGF was interpreted to become input to our ABM simulations. A description of the system evolution and the impact assessment highlighting the benefit in using an ABM simulation for assessing the impact of new technologies on the electricity distribution grid are finally given.

8.5.1 Description of four simulations derived from scenario "Rise of the Prosumer" of the Future Grid Forum

The Future Grid Forum was created in 2012 with 120 representatives of every segment of the electricity industry, government and community, to study and review the drivers of change in the electricity sector, in a whole-of-system approach for Australia. The outcome of the FGF consists of a report that describes 4 scenarios with distinct options, which are not mutually exclusive. The economic impact they will have as well as which sectors are most likely to be impacted by each of these is also given. Based on their output, we chose to use scenario B as input to our ABM.

This scenario, called "Rise of the Prosumer", expects that customers will be more involved in the design of their product for their own needs. It was translated in terms of technology in the electricity sector to a rise of on-site generation of electricity, using rooftop photovoltaic (reaching a capacity of 46% by 2050), and an increase in the number of electric vehicles (reaching 27% of the vehicle fleet by 2050). This shift in attitude and technology uptake is expected to lead to a decline in centralised power generation and a rise of the importance of a distributed system, where the customer is at the centre of the system, where they consume, trade, generate and store electricity.

This scenario was chosen for our case study because of its decentralised nature in terms of technology uptake, and the fact that individuals' behaviours in using these technologies might cause unexpected outcomes at the system level. The strength of the agent-based model lies in the fact that it is capable of taking into account individuals' decisions (the agents), which are here the customers behaviours, and seeing what might happen at the system level (the network or parts of it) when they all are acting and interacting with one another. Finally, because these are new technologies for which we have no data to draw from to understand what their impact might be, they suit very well a need to be modelled using ABM.

While scenario B only has one set of hypotheses, the impact of it unfolding might have different consequences depending on where it was to take place, as well as how these technologies might be used. As such, we created four simulations from it, where two different networks, and two types of charging methods, which could be the results of a policy, are considered.

Simulations were done over two areas in order to understand the geographical implication of a variation in impact of the same behaviour. This illustrates the fact that the location of uptake of a technology in the network matters, as in some areas drastic changes might be observed, while another area might not be affected at all. The two areas that were selected in Townsville have distinct characteristics in terms of load profiles:

- **Townsville Central** has its load dominated by commercial customers (residential premises represent 26.6% of the load, commercial ones **67.6%**, and industrial ones 5.8%);
- **Townsville Residential** has its load dominated by residential customers (residential premises represent **72.8%** of the load, commercial ones 25.1%, and industrial ones 2.1%)

Simulations were also run when changing the charging methods for the electric vehicles which might have a different impact on the peak load at each of these two locations:

- **Uncontrolled charging** of EVs, where the EVs start charging as soon as the vehicle gets to the premise;
- **Controlled charging** of EVs, where the charging will start only between 8pm and midnight.

In both methods, it was assumed that the EV battery would be used for charging only, after the end of the vehicle trips for each day, and that the vehicles would charge through the night until the battery is full. The constraint on controlled charging start time (8pm to midnight) was chosen such that additional load to the peak load (between 4 and 8pm) would be avoided, and that charging would finish before 7am the following day, ensuring that EV users would have a full battery before starting their activities.

In summary, from the initial scenario B of the FGF, four cases were further developed to capture the spatio-temporal characteristics of electricity consumption at a fine level of detail. Where and how the technologies are being used might impact the network in very different ways, leading to different planning and management requirements that the distribution network providers need to be aware of.

8.5.2 System Description - Setting up the simulations

Setting up these four simulations was done using the plugins provided by MODAM. Different plugins and data providers were required to set up the agent-based model, based on the descriptions from Table 8-2 and Table 8-3. These can be summarised as follows:

- **Baseline Information for Townsville**
    - **Baseline network information** describing the assets on the network (transformers, switches, lines …) and the premises (of type residential, commercial, industrial) and how they are related to one another (topology)
    - **Baseline load at individual premises** that are informing the behaviours of the agents using three input parameters
        - Half-hourly interval demand curves chosen using a combination of historical records of individual premises, feeder profiles scaled to premise electricity consumption, and a weather-dependent model. This combination of load data will vary from one network to another, depending on the available data and their characteristics;
        - Yearly overall electricity consumption for each premise over the network;
        - Yearly growth rates. Overall demand growth was modelled using the expected peaks loads at 10 PoE as reported in (Ergon

Energy, 2013b) for the years 2013-2019, and extrapolated for the remainder years of the simulation period.

- **Scenario specific information**
  - **Rooftop PV**
    - The rooftop PVs were added to the base network, where individual devices were assigned to premises according to rates of uptake and socio-demographic characteristics at the premise location. A large number of PV types (with varying size, tilt and azimuth) were created with the view of capturing the diversity in the premise orientations and system installations over Townsville;
    - the half-hourly output for each system was calculated using a weather-dependent model where temperature and cloud passage over the device was modelled, as described in (Boulaire, Utting, Drogemuller, Abeygunawardana, et al., 2012).
  - **Electric Vehicle**
    - Electric vehicles were added to the base network, where individual devices were also assigned to premises according to rates of uptake and socio-demographic characteristics of the premise location. Two types of EVs were considered: a plugin-hybrid (PHEV) with a capacity of 16kWh and a Battery EV (BEV) with a capacity of 25kWh;
    - Time-based load for each system was calculated using a probability function on time of arrival at the premise and state of charge prior to charging. Data on home arrivals were extracted from (Verdant Vision, 2012), using the Verdant data only for the case of uncontrolled charging. In the case of controlled charging, the start-charging times were chosen randomly, spread between 8pm and midnight. Data on state of charge were extracted from (Smart Grid Smart City, 2012) which had two datasets, one for residential fleet charging and one for commercial fleet charging. For both these parameters,

probabilities for different times of arrival and state of charge were given. These were used to build a cumulative density function from which a value at each new day was obtained by applying the roulette wheel selection method.

    - The charging power was set at 3.6kW, being the level 1 for EV charging as described in (AEMC, 2012) which corresponds to a residential charging station (15 amps).

- Weather information - weather data are used in a few of the models (baseline load, rooftop PV output) and influences the behaviours over time (seasons). Historic weather patterns for Townsville were replayed during the simulation to ensure realistic variations.

A description of the creation of the input data specific to the scenario is given below. Pre-processing of data was performed and became the input to the model, which defined how the PV and EV allocation was done over the distribution network.

8.5.3 Exogenous scenarios as input to the simulations - temporal and geospatial allocation of rooftop PVs and EVs over the network

Two types of exogenous parameters from scenario B of the FGF were required. These parameters, which are models of uptake of technology over a certain region, were pre-processed and used as input to the simulation.

Understanding where EVs are likely to be recharged is important from the point of view of the grid operator as they might have more or less impact on the planning of the grid. Indeed, if many EVs are purchased by people living in one area connected to the same distribution transformer, this transformer might become overloaded at times. This might lead to a need for network upgrade for its safe and reliable operation. However, if the EVs are uniformly distributed over the network, while charging might create a new distinct pattern in the daily or weekly load, their impact might not be as problematic for the distribution network. The same is true for PVs which might create rises in voltages if the output exceeds the consumption requirements in a low voltage network.

Temporal and spatial allocation of rooftop PVs to premises over Townsville

The temporal and spatial allocation of rooftop PVs to premises over Townsville was done following three steps:

1. Creation of a map of the expected numbers of installations of rooftop PVs in Townsville at the SA1 level (Statistical Area level 1) for each year over the 2012-2050 period.
    - An uptake curve of rooftop PV for Townsville was created using actual data extracted from (Clean Energy Regulator, 2014b) and forecasts for rooftop PV installed capacity from (AEMO, 2012) for Queensland.
    - These uptake rates were then applied to Townsville's statistical areas level 1 (SA1) as defined by the Australian Bureau of Statistics (Australian Bureau of Statistics, 2010) using criteria regarding the housing types. Further, using saturation criteria from which the uptake curves were derived (AEMO, 2012), the number of candidate houses for PV installations in each SA1 was calculated as "the number of occupied, detached houses, plus 30% of other dwelling types" multiplied by the percentage calculated in the uptake curves. Additional allowances were made for commercial premises from our data, where 30% of commercial premises became candidates for PV installations.

Because the number of occupied detached houses varies from one SA1 to another, allocation of PVs following this criterion creates diversity across Townsville. Figure 8-1 shows two maps of installed rooftop PVs over the SA1s for 2020 and 2032. The different colours indicate the range of rooftop PVs that are expected to be installed over time, showing the diversity over space.

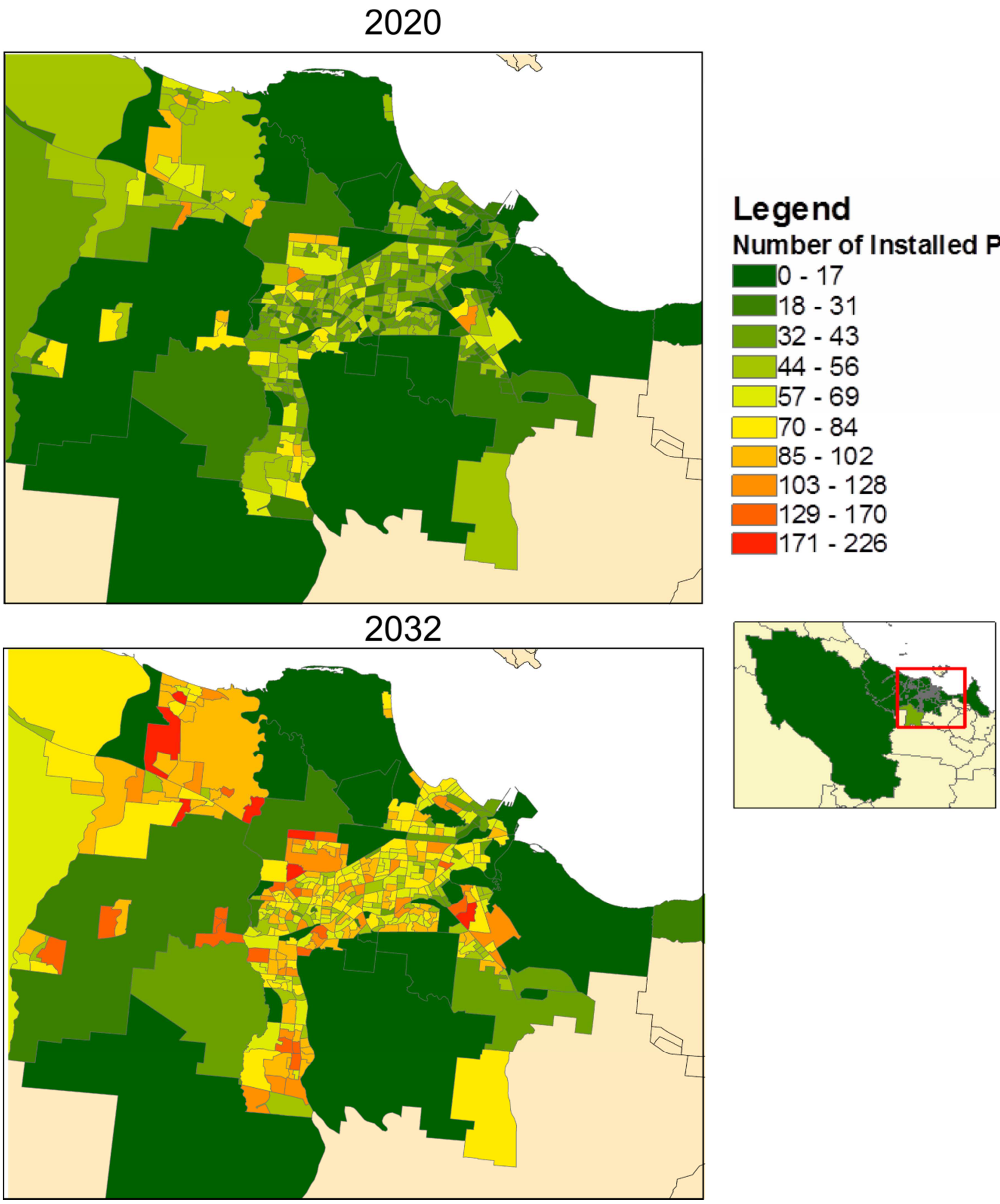


Figure 8-1 - Expected Number of Rooftop PV Installations for Townsville for 2020 and 2032 at the SA1 level

2. Creation of the PV systems characteristics to be assigned to the premises over Townsville. A file of 1,000 rooftop PVs was created with the view of capturing the diversity in the premises orientations and systems installations.
   - PVs were created with varying values for their DC rating, their array tilt and azimuth which were sampled from distribution functions created from

datasets using (Clean Energy Regulator, 2014a), (National Renewable Energy Laboratory, 2012) and (National Renewable Energy Laboratory, 2014).

3. Allocation of PV systems to premises within each SA1 over the 2012-2050 period.
   - Within each SA1, premises described in the topology file from Ergon were chosen randomly to match the expected number of PVs to be installed in each year. To those selected premises, a PV asset from the 1,000 different types created was allocated in a semi-random manner. The selection of the PV was done on its size influenced by the energy consumption of the premise, known from the annual billing data. The allocation of the PVs to the premises was done by separating the billing information into 4 groups of consumption, as well as creating four groups of PV installations according to their size. This followed the assumption that premises with higher electricity consumption, which are often those that are the biggest in size, will more likely have the biggest size solar panels (to offset their usage from the grid, but also because they are more likely to have more roof space on which to install more solar panels). While the pool of PVs for residential premises covered the four categories, those for commercial premises were limited to the class with the largest PV sizes.

From these calculations, the input to the ABM simulation was a dataset containing the information about which PV was installed to which premise in which year over the simulation period.

Temporal and spatial allocation of Electric Vehicles to premises over Townsville

In a similar manner to the rooftop PV systems uptake, the input to the simulation for electric vehicles was created using a wide range of resources, and following three steps:

1. Creation of a map of the expected numbers of EVs in service in Townsville at the SA1 level for each year over the 2012-2050 period.

- The total number of EVs in service for each year for the whole area of Townsville was calculated. Take-up rates of EVs in Queensland were calculated, as a percentage of total sales using the predicted number of EV (BEV and PHEV) sales and all vehicle sales for Queensland based on the central scenario from (AECOM Australia Pty Ltd, 2012), information about demographics (expected population growth for Australia (Queensland Government, 2011)), and expected number of vehicles per 1,000 people in Australia (International Energy Agency, 2014). These rates were then applied to Townsville and converted to estimate the number of new EV sales in each year. The rate of vehicle renewal was then taken into account, which was chosen as 10 years, to get the total number of EVs in service in each year. Finally, this was converted to the number of new EVs and the percentage of EVs in service in each year.
- The uptake of the EVs was modelled at the SA1 level using demographics information in the view of capturing those customers who might purchase EVs. From (AECOM Australia Pty Ltd, 2012) it is expected that EV purchases will be observed in spatial clusters in the early years, where take up will most likely happen in urban and major hub areas, followed by an uptake by early adopters "*who are typically characterised as having higher incomes, higher levels of education, and being more technologically and environmentally aware*". Consequently, data about income was extracted from the ABS census data at the SA1 level, so that EVs were assigned first to these areas that are likely to have early adopters. According to Roger's Bell curve that captures the technology adoption lifecycle, early adopters are the first 16% buyers of the market. After 16% of EV uptake, which corresponds to year 2028 in our dataset, it was assumed that anyone, regardless of their income was a candidate to purchase an EV. Different models of uptake were trialled, and one was kept where EVs are taken up firstly in the SA1s that have the highest proportion of high incomes, and limited to the people who have high incomes; high income earners were chosen as those greater than $78,000 a year, which is above the average Australian wage.

This model's map first shows spatial clusters of adoption, where the high income earners happen to be. Then the EV adoption spreads to the high income earners in the SA1s that have a lower proportion of high income earners, until all the SA1s become candidates to receive EVs, as they become widely adopted (43.9% of new sales). The allocation of the EVs over time is shown on Figure 8-2 for 2020 and 2032. The clusters of EV adoptions are initially localised, as indicated by the light green regions in 2020, which also indicates where the high income earners are; then all the SA1s have EVs purchased, but clusters of higher numbers can still be distinguished, with red patches in 2032.

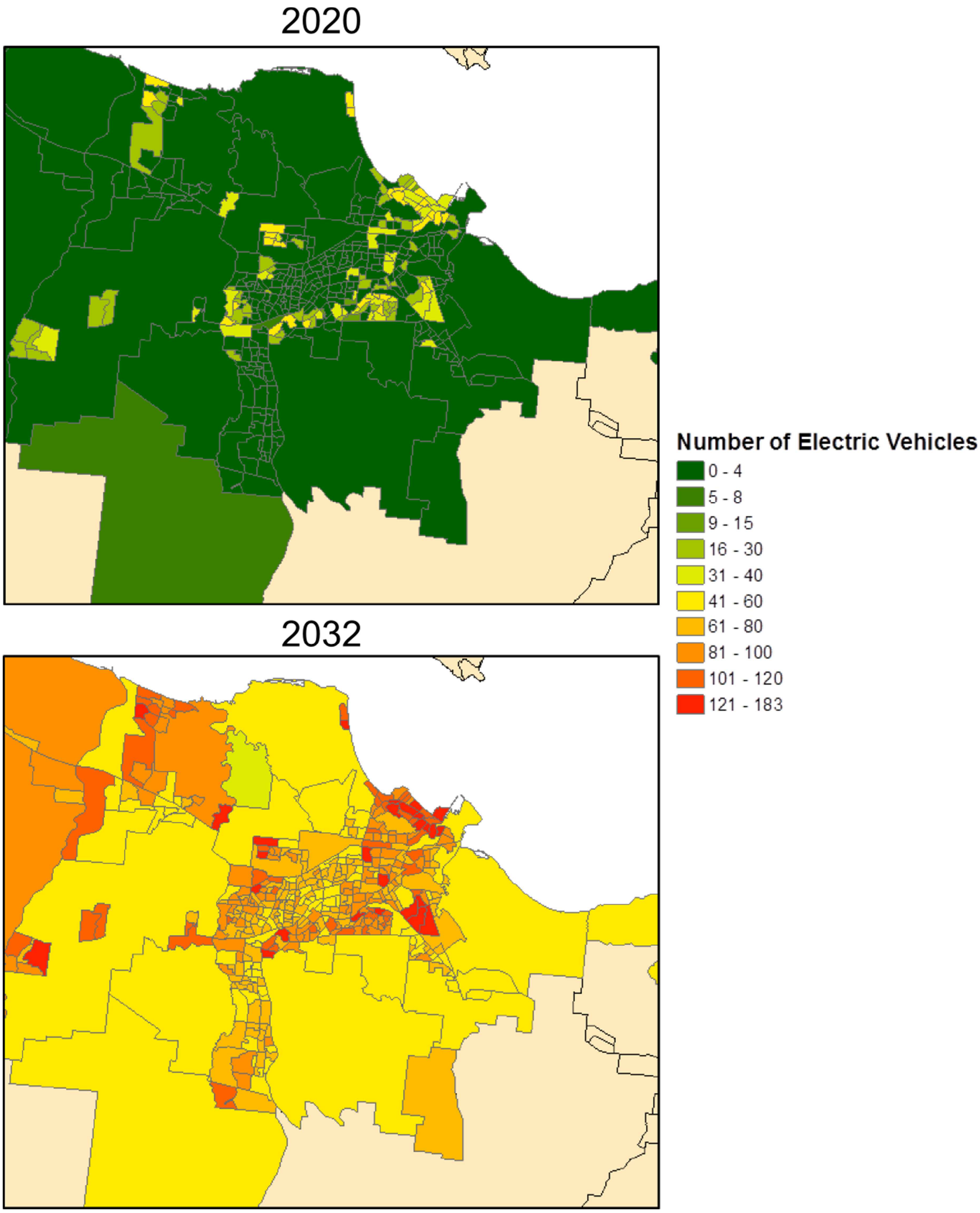


Figure 8-2 - Expected number of electric vehicles in Townsville for 2020 and 2032 at the SA1 level

2. Creation of the EVs characteristics to be assigned to the premises over Townsville. Two types of EVs that are representative of a BEV and a PHEV were considered, similar to those described in (Paevere et al., 2014). One battery was assumed to have a capacity of 25kWh, representative of a BEV, and the other to be 16kWh, representative of a PHEV. For both, the charging power was set at 3.6kW, being the level 1 for EV charging as described in (AEMC, 2012)

which corresponds to a residential charging station. This value was set as the inverter rate for both batteries, to be used in the algorithm for the charging regimes.

3. These two types of batteries were then allocated according to the proportions of expected EVs and PHEVs described in (AECOM Australia Pty Ltd, 2012) in a random manner to the premises that had previously been selected as candidates for an EV according to their income (step 1).

A file was finally produced that contains the allocation of a BEV or PHEV battery to a given number of premises within a SA1 according to their likelihood of having purchased it from their income level. That file was later used as input to the ABM simulation.

8.5.4 System evolution - running the simulations

Four simulations were then run with the models for uptake of EV and PV described above as input. These simulations were done for Townsville over the 2013-2032 period.

As the simulations were done over two distinct zones in Townsville, the number of agents varied from one simulation to the other. For Townsville Central, 16,365 premises were simulated using three modes of load where 65% of the load was based on historical individual premise load profiles, 10% of the load was simulated using feeder historical data and 25% using the weather-dependent model. This choice of model weight was based on the types of premises making up this area, so that most residential premises (representing 26.6% of the load) would be using the weather-dependent model that was developed from the analysis of residential consumption data, and the larger portion of the dataset would be using historical data. Finally, 10% of the loads were chosen to be using feeder historical data to provide some uniformity in the final simulated load. For any of these load modelling techniques, each premise had its half-hourly load scaled so that its yearly electricity load corresponded to the one reported in the billing information as supplied by Ergon Energy. A growth factor was also applied to these loads according to the expected average and peak load growth of each feeder over the simulation period, as provided by Ergon Energy. In addition, some of these premises had a PV and/or an EV influencing their base load. As such, 5,675 EVs and 5,853 PVs were assigned to the premises over the 20 years of the simulation period. In total, this network simulation resulted in 56,288 agents having their load simulated either as it is consumed (e.g. at the premise) or flowing through (e.g. transformers, switches, lines…).

Similarly, for Townsville Residential, 25,797 premise loads were simulated, with 11,009 PVs and 11,987 EVs being added to the network over the years. The simulation resulted in 100,428 agents, including premises, PV and EV agents, as well as the different network agents (lines, switches, transformers, buses).

The simulations were run as a sequential program on an i7-2620M CPU, and took between 20 and 75 minutes per simulated year, depending on the network size and the EV charging method.

Outputs of the simulations were saved in csv files for various nodes of interest in the network for each of the years of the simulation period. As an example, the simulated data was aggregated for each of the areas of interest, as well as for each of the zone substations as described in the Ergon files, for the peak week and for each ½ hourly timestamp over the simulation period. Half-hourly load for each transformer was also saved to investigate the variation on the load over space and time. This data was used first for the model performance assessment discussed below, as well as the simulations assessments.

8.5.5 Model performance assessment

As mentioned in section 8.4, the different modules of the model have been subject to verification and validation with an initial set of data. However, because MODAM allows different types of input data, and a large number of modules combinations, these might influence greatly the output of the simulations and might lead to unreliable outcomes. Consequently, assessment of the model performance is required for each of the created models using MODAM. Because of the setup of the simulations described above, two types of validation were required: validation of the input data to the ABM model (exogenous model) and validation of the output of the ABM simulations. These were done following the different steps described in (Bennett et al., 2013).

The validation of the exogenous model was done for PV installations by comparing data from our model to recorded data available from the Clean Energy Regulator website (Clean Energy Regulator, 2014b) at the postcode level. Our model estimated the number of solar panel installation for each year and each SA1, which are contained within a postcode; an aggregation of our model data was then done over a table of SA1-postcode correspondence. The number and capacity of PV installations from our model were compared to the recorded ones for 2012-2014, and showed that our model underestimated the number of PV by 7.8% and capacity by 9.6%. This underestimation might be caused by the household data at the SA1 level from which our model was built, as only data for premises in 2011, which is the year of the census, was used. Townsville being a growing city, the number of newly built houses has increased over the last few years, with a majority of newly-built premises being individual houses, which would have been candidates for PV installation in our model. Despite this difference in results, this model was considered reasonable, keeping in mind that improvements for its next version will include newly built-houses.

For EVs, however, it was not possible to validate the model with actual records because of the lack of data due to the still very limited take-up of these technologies.

In both cases, the input data for EV and PV will gradually be replaced by actual data as they become available, while projections for future uptake will be refined over time with new models. Such update to our model is facilitated thanks to the modular approach to our implementation.

Validation of our ABM was done for the first year of our simulations which is the benchmark year, the other years being projections. This was done by comparing the state variable from the simulation to recorded data at the feeder level. While some of this feeder data is used for the calibration of the model, its ratio is of 10% to the rest of the data which is reasonable for meaningful comparison. Different metrics were used, which compared the simulated chronological demand curves and the load duration curves (curve of loads over a year ordered in decreasing order) aggregated at the feeder level with actual measurements for the same feeders, as well as their peak load.

Figure 8-3 shows the simulated and actual load for one feeder over the month of January as a chronological curve, and the load duration curve for the whole year, for that same feeder. The degree of accuracy varies over the month in the chronological curve; however, the shape within the days and the weeks is respected, with the weekends having lower consumption intensity than the weekdays. While some peaks days during the month of January haven't been captured, the load duration curve shows that the general shape of the load has been quite well estimated. The difference between the model peak load and the actual one resulted in an underestimation of 11% from our model.

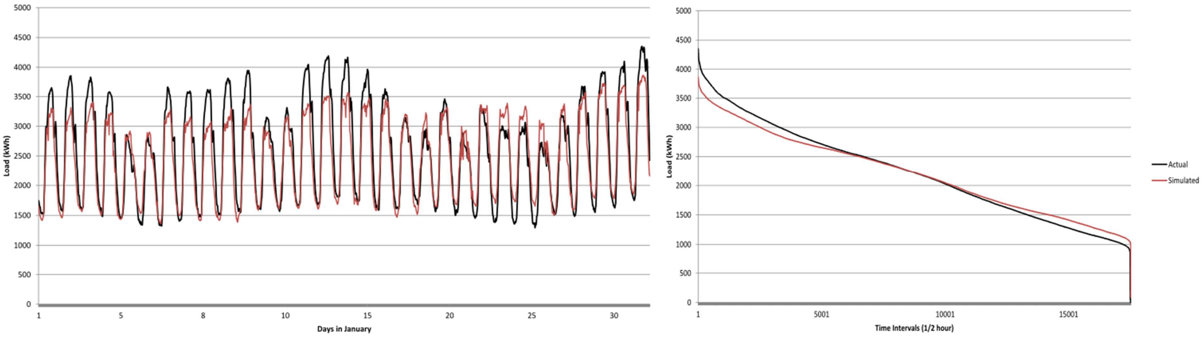


Figure 8-3 - Comparison of load data measured at a given feeder and simulation output - chronological and load duration curves

This type of validation was undertaken for all the feeders over the area of study, which highlighted problems in some of the data, with missing information about premises attached to a feeder for some areas, as well as unexpected changes in the collected data when load has been shifted from one feeder to another for example, for management reasons. Overall, our simulations underestimated the peak load by 33% for the benchmark year. While this variable shows a larger difference than hoped between the simulated and observed data, considering the difficulty in predicting peak consumption because of the many factors causing it, it was considered acceptable for now. Further, for future years in the simulation, a growth factor was added to the model to adjust the peak load. These were compared with the modelled peak load from Ergon Energy and differed by 2%, which is reasonable.

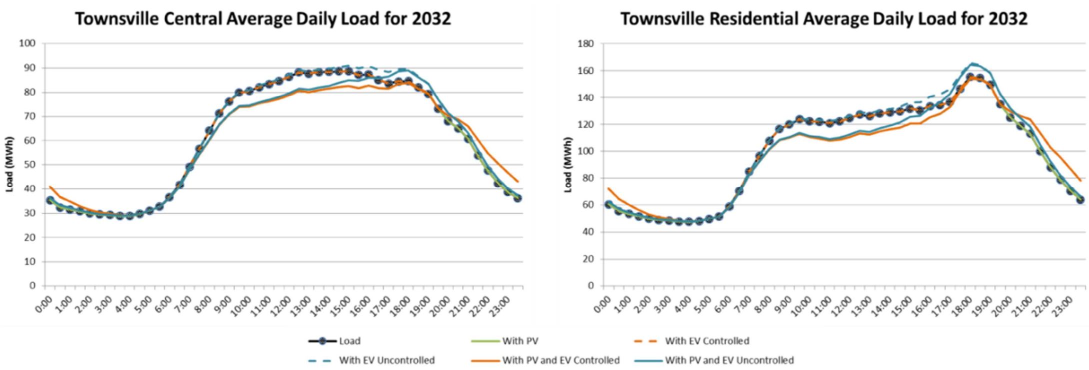
Overall, while it would be beneficial to obtain more reliable input data as well as further refine the model to obtain more accurate peak load predictions, the current model has allowed understanding the current network better. Further, it allows having a discussion about the different areas of concerns in one single picture where the interactions of the different components have been taken into account. From this broad view picture, areas of interest can be further investigated by the planners.

8.5.6 Impact assessment of the simulations

Large amounts of data were produced by the simulations, enabling many different assessments of the impact of scenario B of the FGF to be done. This section describes a few of these results, highlighting the capacity of these agent-based simulations to provide an understanding of what might happen over a system as a whole as it evolves, as well as at a fine level of detail over time and over space.

Impact assessment at the system level

The simulation outputs provide overall trends over the years at the system level which can be useful to analyse in order to understand how the system as a whole has evolved. As an illustration, Figure 8-4 shows the average daily load for each network in 2032, under the two EV charging methods. We can see the very distinct patterns of a commercial load and a residential load, which are dominating these areas. Also, the influence of the high penetration rate of PV (47%) and EV (19%) in 2032 is noticeable. The commercial load, whose pattern has the characteristic of being flat during the day, has had its shape slightly modified. The load has been reduced during the day in a flat manner thanks to the PVs, consequently lowering the peak load; however, a peak has started to appear in the evening as a greater number of electric vehicles are starting to charge around 6pm, in the case of uncontrolled charging. The residential load however has kept a similar pattern to its base load pattern (with and without these technologies), but the evening peak has been accentuated with the addition of EVs, in the case of uncontrolled charging.



(a) (b)

Figure 8-4 - Daily load overage over a zone with a predominantly commercial load (a), and a residential zone (b)

Graphs like these are interesting to understand the overall behaviour of these technologies over a year and for the whole system under study and can provide different types of useful information. As an illustration, they can also be used to calculate the greenhouse gas emission reductions when a certain technology uptake increases, as the overall electricity consumption over a year can be calculated.

For other purposes, however, such as when planning a distribution network, the metric of interest is the peak load. Because we are interested in this purpose, the peak load was further analysed and discussed in the rest of this paper.

Peak Load

Similarly to the previous example, the peak load was investigated over both areas for each of the years. An example of the peak day for Townsville Central is shown in Figure 8-5 for three years that have increasing percentage of PV and EV uptake, comparing the two charging methods for EVs. Because these graphs are peak days and not an average, and because they are derived from actual records, they can show at times differing patterns, as is the case for example for 2030 which shows a sudden drop around 10am. This might be due to a sudden interruption in service delivery over a small area of the zone for example, as captured in the historical data, which could also happen in the future.

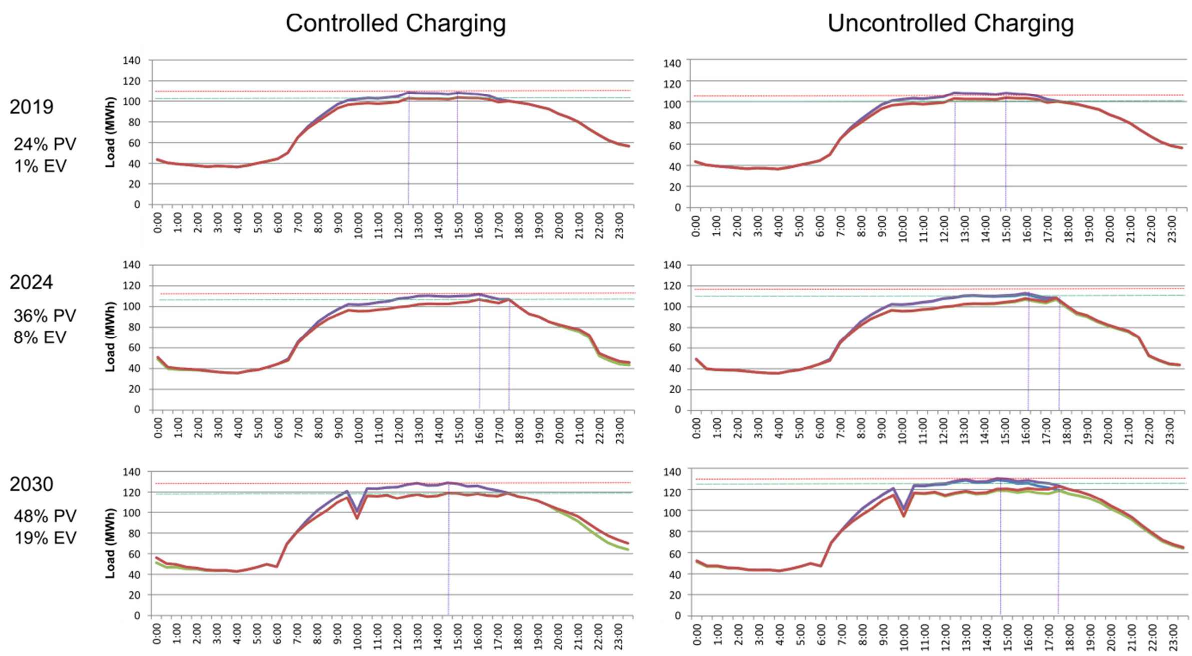


Figure 8-5 - Peak load over Townsville Central Zone simulated for 2019, 2024 and 2032.

In Figure 8-5, we can see that regardless of the charging method we have a reduction in the peak load for Townsville Central over all the years of the simulation period. A year of interest is 2030 which is starting to show EVs having an impact on the peak load with uncontrolled charging. While the peak is reduced with PV and EV

(-6.63% of the base load, -5MWh over 30 minutes) , the benefit of PV is starting to be reduced as EV charging is starting to impact the peak time (+1.1%), causing the peak on that day, and displacing its time from 2.30pm to 5.30pm.

Having the EVs on the network using the uncontrolled charging method will limit the benefit of the PVs in reducing the peak. This is important to understand when each of the technologies are going to stop being beneficial in reaching a goal. In this case, it shows that increasing the percentage of PV on the network is beneficial for this area, until EVs are starting to reach higher percentages (19% of the fleet) and if there is no policy in place to constrain their charging to different times of the day. As such, incentives could be put in place to increase PV installations up to a certain percentage as long as another policy ensures that charging of EVs will only happen over certain periods during the day.

While the shift in peak happened on the same day in the examples given in Figure 8-5, there are years for which the peak was not only shifted to a different time of the day but also a different time of the year. Understanding how far the peak can be reduced thanks to a policy or a technology is important so that their limitations can be understood. Because the simulations are done at a fine level of detail over time, it is possible to identify those times when another technology will limit the promised reduction of the first one.

Similar graphs were also plotted for Townsville Residential, but are not shown here. A reduction in the peak load was also observed in all the years when using the controlled charging method. However, for the uncontrolled charging method, EVs impact was noticeable at lower percentages of EV penetration (in 2019, with only 2% of EV). In that case, the time of peak was not shifted to another time, but the peak load intensity was aggravated by the addition of the EV charging load. Depending on the nature of the system under study, different effects of the technologies can be observed.

Further to these, graphs of the changes in peak load were drawn, Figure 8-6. These capture how the peak load over each area varies depending on the impact of the two technologies added (PV and EV), and the method of charging used for EV. In all these graphs, the percentage of variation from the base load for both technologies is given: in green is the reduction in peak load due to installations of

PV, and in purple is the percentage of load increase due to EVs. The combination of the impact of the two technologies on the peak load is shown in red.

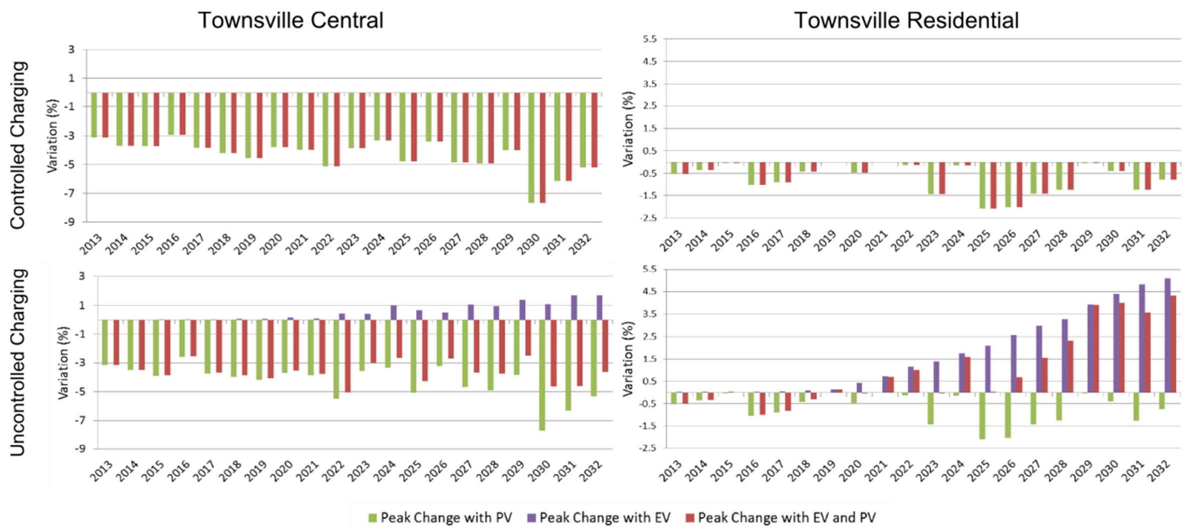


Figure 8-6 - Variation in load when the Zone peaks - Shifts in peak with addition of new technologies over the years: PV and EV, and when the electric vehicles have controlled and uncontrolled charging methods, over Townsville Central and Townsville Residential.

We can see in Figure 8-6 that for Townsville Central, the peak of the zone has decreased, whether the EV charging method is controlled or uncontrolled, thanks to the presence of PVs. For controlled charging the peak changes by an average of -4.3% over all the years of the simulation (ranging between -2.9% and -7.6%) and for uncontrolled charging it changes by -3.4% (between -2.4% and -5%). With the uncontrolled charging method, as the percentage of EVs increases over the network over the years, their contribution to the load is also noticeable. This is caused by the charging being allowed to happen at any time of the day when an EV gets to its premise and starts charging. In addition, the decrease in load thanks to the PVs contribution is constrained by the shift in time of the peak, which might now happen at 6.30pm. However, while the EV peaks are starting to be noticeable (+1.7% in 2031-32), their effect is sufficiently minor that overall, a decrease in peak load is still observed thanks to PVs.

Different behaviours can be observed however for Townsville Residential. The peak load is always decreased over the simulation period with the controlled charging, but the variation in the decrease is much lower than the one observed for Townsville Central (average of -1.1% over all the years of the simulation). In some years, the change is very close to 0 as the PV contribution is not very strong, because the peak is happening around sunset. Further, because the time of the peak is between 5pm and 8pm for the residential zone, as soon as the EV charging method is switched to uncontrolled charging, the EV contribution to the load leads to an increase of the overall peak load at the zone. One important point here is that the influence of EVs on the peak load is happening at very low percentages of EV penetration. Indeed, in 2019, we have the first increase in the peak load caused by EV, however small (0.13%) and also because the PV contribution could not overcome it. In 2024, similarly, PV output has not helped reduce the peak much on that day (-0.14%), but the load increase due to EV (+1.76%), led to an overall load increase of 1.6%. In 2030, again PV output only reduced the load by 0.4% and EV increased it by 4.4% which means that the overall increase was 4%, which is about an average of 7.5MWh.

In conclusion, many useful insights into the impact of PV and EV can be gained by assessing the results of the simulation outputs over the whole area under study. Thanks to the versatility of our model, these results can be used for different purposes, such as to understand the overall reduction in electricity demand over a year, or the peak demand. Also, understanding how the reduction in peak load is obtained and how the time of peak is shifted gives insight as to how far a given technology is able to help reduce the load for a given case.

Impact assessment at the asset level

While it is interesting to understand how the load is impacted over a whole area, it is also important to understand what happens at the different nodes within the network when the load peaks on individual assets. This peak might happen at a different time to the zone peak, which is important to know as sizing of the equipment by the distribution network planners will depend on the local peak value. In this paper, we have chosen to study the distribution transformers further.

Because the simulation can give the load for each asset for each ½ hour of the simulation period, we were able to identify the peak of each transformer in both areas. In a similar way to the graph above, the variation in load at each transformer peak was calculated for each year, as a percentage of deviation from the base load. These variations were then averaged for each year, and the results are shown in Figure 8-7.

Figure 8-7 - Average variation in peak load over the transformers when adding new technologies (PV and EV), for Townsville Central and Townsville Residential and two charging modes for EVs

Figure 8-7 shows that only in the "Townsville Central with controlled charging" scenario will the transformers see a reduction in their peak on average over all the years of the simulation. In the other three cases, while the EV penetration rate is still low, the average over all the transformers is negative thanks to the contribution of PV; but in the later years of the simulation, the EVs impact the peak of the transformers, leading them to increase. Both Townsville Central and Townsville Residential transformers see an increased peak on average in 2026 when the charging method is uncontrolled. In these cases, the benefit from PV is counteracted by the increase of EV, as it reaches 11.85% of the vehicles in service. For Townsville Residential, with controlled charging, EVs impact the load negatively in 2032 as the EV rate reaches 21.1%.

While these observations are still averages over the transformer peaks, it is important to notice how wide the standard deviation bars are for all of the graphs, and more specifically for Townsville Residential, showing the disparity amongst the transformers within each year. Only partial standard deviation bars were shown for clarity reasons in this paper. But, in both the Townsville Residential scenarios the standard deviations of the peak shift with EV and PV (red bars) gradually decrease from 179% in 2013 to around 70% in 2032, as EVs become more common and more uniformly distributed. This emphasises the importance of understanding the effect of technologies at the network asset level. Indeed, Figure 8-7 shows that even for Townsville Central with controlled charging, there will be transformers that will see their peak increasing, even at low percentages of EV penetration. These observations are even more important for Townsville Residential, which observes much wider variation around the mean because these transformers are peaking around 6pm and 8pm on average.

Understanding this variation in peak at the transformer is important as this is what can create problems on the network, especially as the impact can be localised in terms of time of the day but also location in the network. Similar to the problem of voltage rises that have been observed on some low-voltage networks due to PV outputs during some of the days when load is low, it can be expected that the impact of EV charging will be localised. This would further stress the network if they happened at the time of the currently existing peak load. Identifying the assets on which such problems might occur, and therefore their areas, is helpful to the planners as they can further investigate those areas and make more informed decisions. For this, visual assessment of the impact of the technology on the network has been done and is presented in the following section.

Visual assessment of the impact of a technology on the network

Thanks to the fine level of detail that the simulation output provides, not only over time but also over space, it is possible to visually represent how a simulation output evolves over the different assets over time. Figure 8-8 shows a map of Townsville Central where the expected number of EVs at the SA1 level are drawn as the background layer, and the transformers are represented as the dots on top of these. The variation in the load for each transformer is shown for its peak time using different colours, where blue indicates little or no change in peak load and red indicates a large variation (up to 84%).

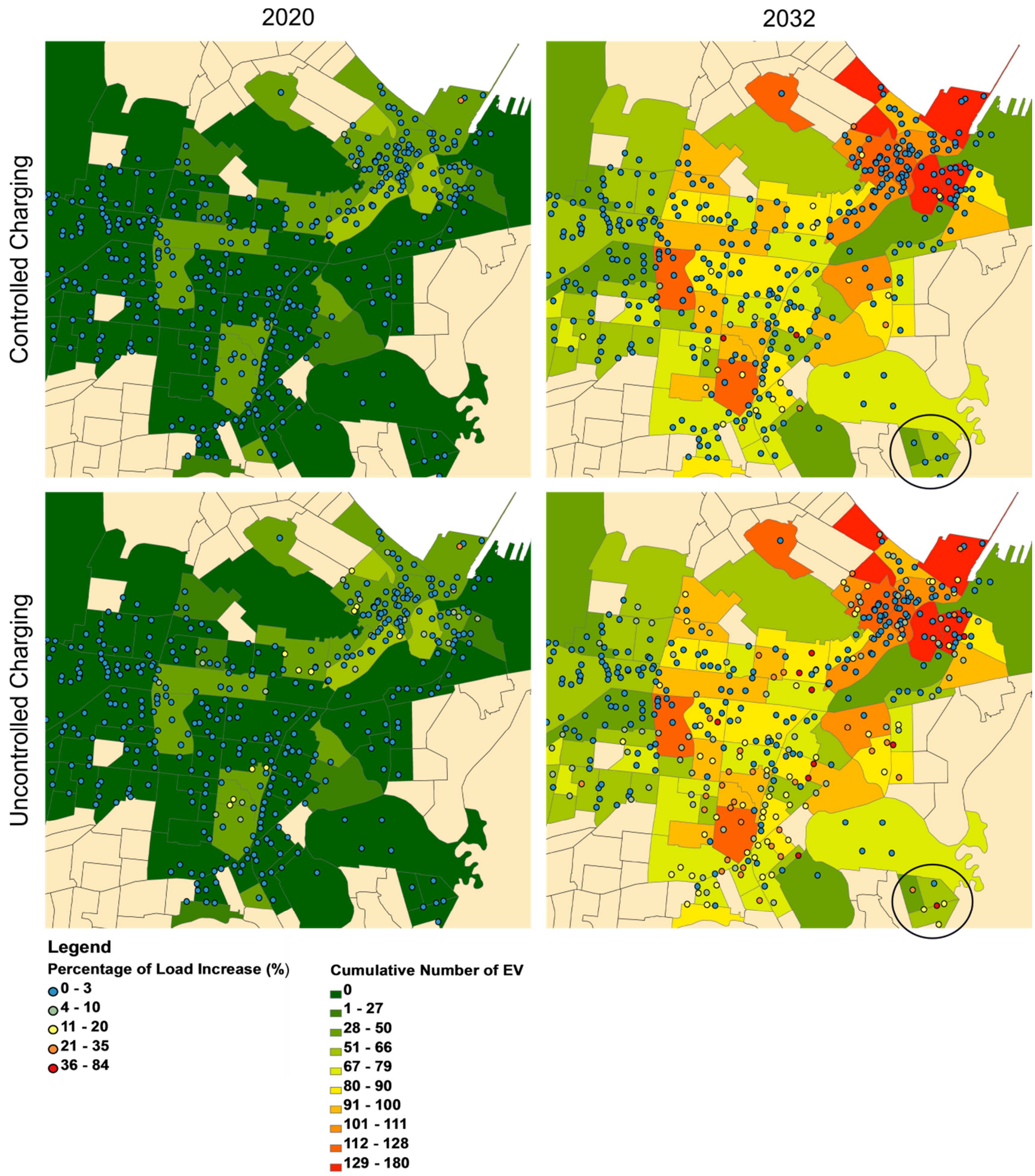


Figure 8-8 - Variation in peak load due to addition of EV on the network for each transformer for Townsville Central for 2020 and 2032

As can be observed in Figure 8-8, in 2020 the percentage of EVs in service is still low (2%), and little or no change in peak load is observed for the controlled charging; however for the uncontrolled charging there are a few transformers that are seeing a shift between 11 and 20% (in yellow). Stronger differences can be observed for 2032 however, when the EVs in service reach 21% of the total fleet in Townsville. Again, fewer changes in peaks are observable for the controlled charging than the uncontrolled charging, but some transformers see a 35% increase in their peak due to EV. For the uncontrolled charging, many more transformers have their peak increased due to PV, and a few reach up to 84% variation.

One interesting thing to note is that, while some transformers are in a zone that has an expected number of EVs that is low to medium, the impact of those few EVs on the grid can still be measurable. An example of this is highlighted by the circles in the graphs in Figure 8-8. In that region, the expected number of EVs is between 28 and 66 in 2032. Depending on the charging method, very different results can be observed. With controlled charging the impact of the EVs is negligible on the peak load (between 0 and 3% increase), however, when using the uncontrolled charging method, the peak of the different transformers has shifted between 11 and 84%, and the 6 transformers in that region are behaving rather differently. This can be explained by the allocation of more EVs on an LV feeder leading to a given transformer, which might happen to have a low baseline peak load that is greatly aggravated by the arrival of EVs.

Finally, these types of observations highlight the importance of modelling the network structure, because different assets (e.g. transformers) can be affected in very different ways, even under the same scenario. Indeed, because of the interrelationship of the other parameters, their effect can be very different at the asset level.

8.6 DISCUSSION

Throughout the impact assessment section, a few points have been highlighted that support the development of an agent-based model to assess the impact of technologies on a distribution network. Thanks to the fine spatial and time representation of the network, and because of the agent-based model nature of the simulation, the following remarks can be made:

- Our system allows taking into account **variability in space**. The spatial influence of the introduction of a technology and/or a policy is represented
    - **The network structure is represented in the model, ensuring the location-specific property of electricity delivery**. The state variable (e.g. the load) is bounded to the physical network as it is in reality, and can better explain the impact a policy might have at a specific location, or over a specific area.

   - **It is possible to analyse the system and draw conclusions at different scales (at the individual level, as well as over regions of aggregation)**. As an example, the simulations in the section above compared the 'load' state variable for two zones of different load nature (Townsville Central and Townsville Residential), while still being able to drill down to the individual transformers. This was further highlighted by the use of GIS mapping of the transformer variations in load. The assets that might have an impact due to the introduction of a given technology can then be identified at a glance and over large areas.

- Our system allows taking into account **variability in time**. The time-dependent characteristics of the electricity delivery is captured through the fine time description of the agents' behaviours
   - **Because peak load is the result of coincidental demand at a given time, the time of use of a new technology is critical to the effect on the peak load**. As a technology is introduced, not only do its place and intensity with which it is used matter, but also the time at which it is used. As long as a technology usage does not coincide with the peak load, increasing its intensity in usage will not affect the peak load, until it has reached such a point that it is creating a new peak (if it is a positive load), or reaching an irreducible point (for a negative load). As an example, increasing the contribution of PV might be useful for commercial zones, but only until the peak load has been reduced to the load value at sunset.
   - **Different usage types of a same technology can be assessed, with the view of informing policy settings**. In addition to the point above, a policy might be better assessed by its limitation resulting from the impact of another load which is irreducible. As an example, for Townsville Central, increasing the amount of PV will not help reducing the peak as long as uncontrolled charging is allowed; however, if controlled charging is chosen, additional PVs would help reduce the peak further.

- Our system allows **information at the individual level to be understood in a global context**
    - **A very large number of agents can be considered at once, with their actions and interactions accounted for**. We have run simulations with more than 100,000 agents over 20 years period at ½ hourly intervals, and with a large number of agents of different nature. While the impact of a technology might be well understood at the individual level, the dynamics of the different behaviours over the system might result in an unexpected outcome at the system level because of the impact of one onto another. For example, expected reductions in the peak load due to the introduction of a technology might be limited by the behaviour of another. This restriction in some cases might even completely void the need for this technology.
    - **The individual impact of a technology as well as the combination of many technologies can be assessed easily**. Results showed the impact of PV and EV individually, on the distribution network of interest, as well as the coincidental impact of the two technologies. Looking at the interrelationship of the two technologies at once is powerful, as while one technology may promise a certain percentage of reduction to the peak, this might not be so, as it might be counteracted by the effect of another technology.
- Our system is **versatile**, allowing the simulation state variable to be used for **different analysis purposes**. For example, the load variable which is an output of the simulation can be used to understand peak load, which is the metric used to design distribution networks; and it is also possible to use it to derive greenhouse gas emissions over a certain area and period when looking at overall load profile over a time period.
- Our system is **flexible** in the way that many simulations can be trialled through different combinations of agents, but also thanks to its capacity to taking input parameters of different format. Not only can variables be set as input to the simulation, but complex input scenarios such as the uptake models of PV and EV presented in the previous section can easily be taken into account.

While these remarks are specific to our application, they can be transferred to other applications in different sectors of the environment, and more specifically to systems that have a networked structure over which agents interact. MODAM is built for large-scale ABMS, and therefore can deal with large infrastructure subject to change (technological or behavioural). This is especially interesting when investigating possible futures over large geographical areas to highlight those that could be at risk in the future and to inform planning decisions.

Examples of such applications are in the water domain as many similarities with the electricity sector can be observed. Indeed, pipes transport water to their points of consumption, and are subject to constraints in their size, connections and flow capacities within the network, similarly to electricity networks. Further, water networks, while still predominantly centralised in Australia, are subject to decentralisation with the introduction of water tanks. These water tanks are very similar in principle to batteries which allow using the resource at different times of the day/year, either to avoid, or reduce reliance on the network. Further, because of the scarcity in water resources in many places and the recent drought over the whole country, policies and technologies have helped changed the way water is consumed in the recent years. Such behaviours have had an impact on the network overall with some areas having a greater impact than others. MODAM would be a possible platform to create scenarios of possible future to understand how water networks might evolve. While MODAM is currently implemented for the electricity sector it can be extended with ease to other domains (water as mentioned above but also gas) thanks to its modular architecture. New assets and agents would be added to the core model, and their behaviour implemented, keeping in mind the specificity of each of the systems.

Finally, from the experience in building our ABMS a few lessons have been learnt regarding different aspects that are applicable to other sectors:

- In a technically changing environment, having mechanisms in place that enable building the model in a flexible manner greatly facilitates the modelling task and allows it to be expanded, within but also beyond the project timeline. While this is not the main contribution of this paper, simulations that were performed and presented here were facilitated by the way the software environment was built, that is using a modular approach to building our ABMS.

- Flexibility in the types of input data to the model is very important. As shown in this paper, the input data to our ABMS can either be raw data or data derived from an exogenous model. Flexibility in the type of input has the advantage to allow for different scenarios to be trialled that can also be quite complex, or to take data of the existing conditions, or using a mix of both.
- Developing our model based on scenarios previously developed by another organisation, such as the FGF in our case, facilitated the discussion with the project partners regarding the use of the model. This created enthusiasm in the partners especially as they had been involved in the initial development of the scenarios and could see how these might impact differently their infrastructure. This also gave confidence in the type of simulations performed.
- Visual assessment of the simulation results is also of great value, especially for communication purposes. Using GIS applications to display output of the ABM brings to the light the geographical component of the simulation, as well as its temporal component when the simulations are played over time. However, while this was possible, due to the large amount of data, some displays were limited to shorter timeframes or smaller areas which is regrettable.
- Validation of the model is still a challenging task, especially with such large and complex systems. However, thanks to better access to data, and especially because of the networked structure of the system, validation was possible. While some metrics, such as the peak load, showed an under prediction by the model, they also highlighted problems in the input data which was valuable to the overall modelling process.

8.7 CONCLUSION AND FUTURE WORK

This paper demonstrated how to use MODAM, an agent-based model platform that allows creating large-scale agent-based models in a fast and simple manner by assembling different components together. This was done through the implementation of one scenario from which four simulations were derived to represent the possible trajectories it might take. These simulations were run over two areas in Townsville, Australia, and highlighted the geographical implications of a variation in impact of a similar behaviour. In addition, two charging methods for the implementation of electric vehicles were investigated, to highlight how the way a technology is used might impact the state variable (e.g. the peak load) at each of these locations.

Thanks to the fine spatio-temporal description of the model by the agent-based model, different insights can be gained at different levels of details. The impact of electric vehicles to the peak load varies depending on the nature of the load for the zone as well as the charging method: with a percentage of EV representing 19% of the vehicle fleet in 2032, a peak increase of 1.7% is observable due to EV in the commercial dominated area, compared to 5.1% for the residential area with uncontrolled charging. The increase in the number of PV could reduce the peak for the commercial area. However, with an uncontrolled charging method, even at a very low percentage of EV (2% of the fleet), the peak load actually increases, because the EV load occurs outside of the PV output hours. When investigating the impact of the technologies at the distribution transformer level, which is the level at which distribution network planners study, wider differences can be noticed. Depending on the charging method and the location, transformers can see an increase up to 84% in their peak load over the commercial area, and increases are even larger over the residential area (over 100%). Observation of these variations over space was made possible by mapping these loads on a GIS map.

The impact assessment of the simulations highlighted a few of the benefits our platform brings to the analysis of the load modelling when modelling technological change in infrastructure. These include: 1) realistic representation of the location-specific property of electricity delivery thanks to the representation of the network structure in the model, which can be analysed at the individual level or aggregated over large areas, 2) ability to capture time-dependent properties of technologies' usage, ensuring that coincidental usage is captured and peak load analysis is realistic, 3) information at the individual level can be understood in a global context, 4) inputs of the simulation can be informed from other models and outputs can be used for further analyses and different purposes.

Finally, because this platform was built with extensibility and flexibility in mind, many simulations can be set up easily. Also many other technologies and behaviours can be added easily in order to extend the models and try many more scenarios. Finally, MODAM can further be extended to other domains that represent agents behaving over a networked structure, such as water or gas networks.

Future work includes extending further the model by adding more agents of technologies or policies that might impact the electricity network and automating the impact assessment of simulation output so that many simulations can be compared on the fly. MODAM can be downloaded for interested users.

8.8 ACKNOWLEDGMENTS

The authors gratefully acknowledge the funding through the NIRAP (National and International Research Alliance Program) grant funded by the Queensland Government and Ergon Energy, which is making this research possible. Also, the contributions of diverse partners on this project, and especially Ergon Energy for providing the data used in the MODAM framework has made this paper possible.

8.9 REFERENCES

Argonne National Laboratory. (2014). The Repast Suite. Retrieved 11/11/2014, 2014, from http://repast.sourceforge.net/

Bae, J. W., Lee, G., & Moon, I.-C. (2012, 2012). *Formal specification supporting incremental and flexible agent-based modeling.* Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.

Batten, D. F., & Grozev, G. (2006). NEMSIM: Finding Ways to Reduce Greenhouse Gas Emissions Using Multi-Agent Electricity Modelling *Complex science for a complex world: exploring human ecosystems with agents* (pp. 227-252). Canberra: ANU E Press.

Bellifemine, F., Caire, G., Trucco, T., & Rimassa, G. (2010). JADE Programmer's Guide. Boston, MA , USA.

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Hoboken, New Jersey, USA: Wiley-Blackwell.

Berryman, M. (2008). *Review of Software Platforms for Agent Based Models*. (DSTO-GD-0532). Edinburgh, South Australia, Australia: Defence Science Technology Organisation.

Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America, 99 Suppl 3*(3), 7280-7287. doi: 10.1073/pnas.082080899

Boulaire, F., Utting, M., & Drogemuller, R. (2013a, 18-26 May 2013). *MODAM: A MODular Agent-based Modelling Framework.* Paper presented at the 2nd International Workshop on Software Engineering Challenges for the Smart Grid as part of the 35th International Conference on Software Engineering (ICSE 2013), San Fransisco, CA, USA.

Boulaire, F., Utting, M., & Drogemuller, R. (2013b, 26/08/2013). *Parallel ABM for electricity distribution grids: a case study.* Paper presented at the 1st Workshop on Parallel and Distributed Agent-Based Simulations, Euro-Par 2013, Aachen, Germany.

Boulaire, F., Utting, M., Drogemuller, R., Abeygunawardana, A., Ledwich, G., & Bell, J. (2012, 6-7 December 2012). *Planning for the Impact of Distributed Solar Energy on the Grid.* Paper presented at the 50th Annual Conference, Australian Solar Energy Society (Australian Solar Council), Swinburne University of Technology, Melbourne.

Boulaire, F., Utting, M., Drogemuller, R., Ledwich, G., & Ziari, I. (2012, 9-12 December 2012). *A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network.* Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.

Briot, J.-P., & Meurisse, T. (2006). A Component-based Model of Agent Behaviors for Multi-Agent-Based Simulations. In P. Stone & G. Weiss (Eds.), *Proceedings of the 7th International Workshop on Multi-Agent-Based Simulation (MABS'06), 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'2006)* (pp. 183–190). New York, NY, USA: Association for Computing Machinery (ACM).

Cai, C., Jahangiri, P., Thomas, A. G., Zhao, H., Aliprantis, D. C., & Tesfatsion, L. (2011, 24-29/07/2011). *Agent-Based Simulation of Distribution Systems with High Penetration of Photovoltaic Generation*, San Diego, CA.

Castiglione, F. (2006). Agent based modeling. *Scholarpedia, 1*(10), 1562. doi: doi::10.4249/scholarpedia.1562

Collier, N. (2013). Repast HPC Manual. Retrieved 01/09/2014, 2014, from http://repast.sourceforge.net/docs/RepastHPCManual.pdf

Collier, N. (2014). Interface ContextBuilder<T>. Retrieved 03/01/2015, 2015, from http://repast.sourceforge.net/docs/api/repast_simphony/index.html

Cordasco, G., Chiara, R. D., Raia, F., Scarano, V., Spagnuolo, C., & Vicidomini, L. (2013). *Designing computational steering facilities for distributed agent based simulations*. Paper presented at the Proceedings of the 2013 ACM SIGSIM conference on Principles of advanced discrete simulation, Montreal, Quebec, Canada.

del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G. (2008). Particle Swarm Optimization: Basic Concepts, Variants and

Applications in Power Systems. *Ieee Transactions On Evolutionary Computation, 12*(2), 171-195. doi: 10.1109/tevc.2007.896686

Dingsøyr, T., Dybå, T., & Moe, N. (2010). Agile Software Development: An Introduction and Overview. In T. Dingsøyr, T. Dybå & N. B. Moe (Eds.), *Agile Software Development* (pp. 1-13). Berlin, Germany: Springer Berlin Heidelberg.

Ergon Energy. (2013). Corporate profile. Retrieved 02/06/2013, 2013, from https://www.ergon.com.au/about-us/who-we-are/our-company/corporate-profile

Gamma, E. (2009). *Design patterns: elements of reusable object-oriented software*. Boston: Addison-Wesley.

Hamill, L. (2010). Agent-based modelling: The next 15 years. *Journal of Artificial Societies and Social Simulation, 13*(4), 7.

Institute for Energy and Transport. (2014, 18/07/2014). Agent Based Modelling for Smart Grids. Retrieved 20/07/2014, 2014, from http://ses.jrc.ec.europa.eu/agent-based-modelling-smart-grids

Klügl, F., & Bazzan, A. L. C. (2012). Agent-based modeling and simulation. *AI Magazine, 33*(3), 29-40.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). MASON: A Multi-Agent Simulation Environment. *Simulation: Transactions of the society for Modeling and Simulation International, 82*(7), 517-527.

Macal, C. M., & North, M. J. (2006, December 3-6, 2006). *Tutorial On Agent-Based Modeling And Simulation Part 2: How To Model With Agents.* Paper presented at the Winter Simulation Conference, Monterey, California, USA.

Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*(4), 151-162.

Morton, A. (2003, 27-30/09). *A fast 'do-it-yourself' load flow algorithm for power systems with sparse topology*. Paper presented at the AUPEC 2003 Australasian Universities Power Engineering Conference, Christchurch, New Zealand.

Najlis, R., Janssen, M. A., & Parker, D. C. (2001, 04-07/10/2001). *Software Tools and Communication Issues.* Paper presented at the Proceedings of a Special Workshop on Land-Use/Land-Cover Change, Irvine, California.

Nikolai, C., & Madey, G. (2009). Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation, 12*(2), 2.

North, M., Conzelmann, G., Koritarov, V., Macal, C., Thimmapuram, P., & Veselka, T. (2002). *E-laboratories : agent-based modeling of electricity markets*. Paper presented at the American Power Conference, Chicago, IL (US).

North, M. J. (2013). A theoretical formalism for analyzing agent-based models. *Complex Adaptive Systems Modeling, 2*(1), 3-3. doi: 10.1186/2194-3206-2-3

North, M. J., & Macal, C. M. (2007). *Managing Business Complexity*. New York, NY: Oxford University Press.

Parker, J. (2007, 2007). *A flexible, large-scale, distributed agent based epidemic model.* Paper presented at the 2007 Winter Simulation Conference, Washington, DC, USA.

Parry, H. R. (2012). Agent Based Modeling, Large Scale Simulations (pp. 76-87). New York, NY: Springer New York.

Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. *Simulation, 82*(9), 609-623.

Schelling, T. C. (1971). Dynamic Models of Segregation *Journal of Mathematical Sociology, 1*, 143-186.

Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008). *EMF: Eclipse Modeling Framework* Boston, MA, USA: Addison-Wesley Professional.

The Eclipse Foundation. (2012). About the Eclipse Foundation. Retrieved 27/02/2012, 2012, from http://www.eclipse.org/org/

Thomas Stober, & Hansmann, U. (2010). Overview of Agile Software Development *Agile Software Development: Best Practices for Large Software Development Projects* (pp. 35-39). Berlin, Heidelberg: Springer Berlin Heidelberg.

Weidlich, A. (2008). *Engineering interrelated electricity markets: an agent-based computational approach*. Heidelberg: Springer [distributor].

Zeigler, B. P. (1976). *Theory of modelling and simulation*. New York, N.Y: Wiley.

Chapter 9: Discussion and Conclusion

This thesis has proposed a compositional approach to building large-scale agent-based models of electrical flows over a distribution network subject to technological change. A software framework, called MODAM, was developed to support the development of large-scale ABMS so that the models can be extended and modified with ease, and the simulations can be setup quickly by choosing from a large range of options, as well as executed at an efficient speed. A library of building blocks of agents was implemented that capture the physical properties of the distribution network and the technologies it is subject to, as well as the way they behave subject to stimuli from the environment or in response to one another's action. Simulations were performed over distribution networks in Townsville, Australia, to validate and investigate the effects of composition on large-scale agent-based models. These were also used to assess the impact of technological change on electricity flows at different geographical and temporal scales, which was initially the motivation for the development of MODAM, and is of benefit to the electricity domain.

This chapter summarises the key contributions from the work presented in this thesis, discusses the advantages and challenges that the novel approach presented in this thesis offers, and finally proposes possible future work directions.

9.1 SUMMARY OF KEY CONTRIBUTIONS

This thesis has made the following contributions:

1. *Development of an agent-based model of electrical flows over a large distribution network subject to technological change*. This model captures the distribution network's physical properties and the actors' behaviours impacting it. This agent-based model describes actual networks with precision where thousands of agents are represented that have been populated from real data extracted from corporate databases (we have run up to 250,000 agents based on available network data, but there is no limit on the number of

agents that can be run). It provides a realistic representation of the location-specific property of electricity delivery, thanks to the representation of the network structure in the model, which can be analysed at the individual level or aggregated over large areas. Analyses can then be scaled up from site specific to nation-wide planning of decentralised generators. This model offers advantages compared to other models that often use synthetic data or take a "copper-plate" approach, where the electricity consumed or added to the network does not have any network constraint;

2. *Development of a novel method that facilitates building large-scale agent-based models and simulations incrementally, as well as offers many options for agents to be defined using alternative, or combination of, behaviours*. This method is based on modularity and composition and is designed to address the specific features of the study domain: networked structure, new technologies integration, massive database options and need to analyse different aspects of the behaviour of agents under a range of scenarios. For this, *dynamic agent composition* was defined that consists of breaking down an agent as an asset and a set of behaviours, where these aspects are defined separately in the model definition and its implementation. These two aspects then come together at runtime, to form what is traditionally called an agent;

3. *Delivery of a software framework, MODAM, that supports the development of scalable networked structured agent-based models and simulations.* This software eases the extension of the model as more information becomes available by providing an architecture that supports implementing the agents in a modular manner. This can be done by independent authors without the need to access previously written code. MODAM also provides a mechanism to handle a wide range of data types and formats for data used to populate the model, making it flexible in terms of data options. It also offers two types of schedulers (sequential and parallel) which have been implemented in-house. The parallel implementation, based on the structure of the agent-based model, increases the speed of executions;

4. *Delivery of a library of building blocks for agent-based models, within the electricity distribution domain, that a user can choose from to build a large number of simulations*. These building blocks are the components in which

the asset and the behaviour descriptions of the agents, as defined according to the *dynamic agent composition,* are stored. These can be implemented by independent authors, and a user can choose from amongst them the ones to use for their simulation, offering a wide range of options;

5. *Development of an architecture that enables an automated agent composition for large-scale agent-based models*. A *Module Manager* brings the components containing the different aspects describing the agents together at runtime without the need for the user to program. This makes trying different scenarios very easy and efficient as it only requires declaring the components that are to be used;
6. *Simulations of technological uptake over a 20-year horizon to understand its impact on grid assets of interest (e.g. transformers).* These simulations have demonstrated the capability of the agent-based model to capture variability of the load in time and space at specific locations over the distribution grid, from the aggregated effect of individual assets' usages and interactions. These simulations are of benefit for planners to inform their decisions when planning the grid, especially as assets at risk of overload, that might need upgrading or replacing, can easily be identified thanks to the visualisation capabilities of MODAM.

9.2 ADVANTAGES AND CHALLENGES TO BUILDING LARGE-SCALE AGENT-BASED MODELS USING A COMPOSITIONAL APPROACH

Taking a compositional approach to building large scale agent-based modelling and simulation applications has many advantages as discussed in many parts of this thesis. A summary of these is given in terms of:

❖ Advantages when developing the agent-based model

- It allows an agent-based model to be extended easily as the information becomes available. The compositional approach allows building on previously developed aspects of the model without requiring a full revamp of the model but also if needed by simply adding new agents without the need to access previously written code;

- Thanks to the interfaces to support the composition of the agent-based model, adding new agents becomes relatively easy. Time spent on implementing some functionalities of the model was monitored. Depending on the complexity of the logic of the assets and the behaviours, it took between half a day and two days for a programmer to add a new agent, where most of the time was spent in the actual implementation of the rules and logic of the behaviours. Bringing the different interfaces together to have a composable model took less than two hours;
- Taking a compositional approach has the potential to considerably speed up the development of modules, as they can be created in parallel by independent developers and come together via common interfaces;
- The separation of assets and behaviours to describe the agents makes it possible to reuse existing code and combine them to create new agents, bringing flexibility in the model definition. This can be done at runtime and does not require writing any code.

❖ Advantages when building the simulations (i.e. running the agent-based model)

- Simulations can quickly be set up by calling on the components that will define the agents, combining the assets and behaviours' logic. This can be done by a non-programmer through the automation of the agent-based model creation in the Module Manager;
- A large number of simulations can be built in an automated manner, using batch files that can vary the combination of plugins, the input files to populate the model, or the parameters used for the simulation. This is extremely valuable especially when trying different policies, for example, in the form of modified behaviour of some agents, to understand their impact at the system level.

However, there are also some challenges with having a compositional approach to building ABMS applications:

❖ Challenges when developing the agent-based model

- Currently, the agents are developed as needed, and in an ad hoc manner. A programmer needs to be familiar with the existing code in order to avoid replications as well as to be able to extend classes for new assets and behaviours. As the number of agents increases, and when many programmers add new agents, it is expected that this process will not be adequate. A remedy to this would be to create an ontology that wraps around the agent-based structure of the simulation platform. This was not part of this thesis, but could be addressed in future work.

❖ Challenges when building the simulations (i.e. running the agent-based model)

- A programmer that extends a model needs to carefully follow the process to add the different factories as well as the required data providers. This can be difficult to understand at first, as quite a few parameters need to be set;
- It can also be difficult for a new user to know which modules to select when starting building simulations with MODAM. This might deter them from using MODAM if they do not use it frequently enough, and develop a deeper knowledge of it. It is recommended to build a GUI tailored to the needs of the user to reduce this risk, as mentioned in chapter 5. However, this limits the modularity benefits to the options that have been selected in the GUI implementation (e.g. selecting input files and parameters, or a set of modules). Also, if the model expands, the GUI also needs to be updated to reflect the new modules, which will also add to the development time;
- As the number of components that can be used to create simulations grows, it can become very difficult to follow which ones can be put together to create a simulation. In (Page & Opper, 1999), the authors describe some of the challenges of the complexity of composable simulations. They present a formal representation of composability and a formal NP-completeness proof that supports the intuition that identifying suitable compositions from a component repository is an intractable problem. While this thesis aimed at providing a method to building flexible and extensible ABMs, it was not the intention to allow generic selection of components to create simulations. However this challenge has to be kept in mind.

Overall, we believe that while adopting a compositional approach to building large-scale ABMS application has its challenges, the advantages far outweigh them especially as the complexity of the model increases. The advantages were especially felt as we implemented the electric vehicle agents for the simulations described in chapter 8. Indeed, their implementation and the setup of the simulations took a day, which is very little time in respect to the complexity of the problem.

Further, the adoption of the compositional approach has little effect on the speed of the execution of the simulations. While time is taken by the Module Manager to bring the components together to form and populate the agent-based model, the agents are then run in the same way as if a non-compositional approach was taken. Because the ABMS discussed in this thesis are large-scale, with hundreds of thousands of agents executed over long time horizons, this setup time is small in proportion to the rest of the simulation time, which can be sped up using the parallel implementation of the scheduler described in chapter 6.

In conclusion, referring back to the research question that was set at the beginning of this thesis, **we can affirm that a compositional approach to building large-scale agent-based models can support modelling the effects of technological change on an electrical distribution network**.

9.3 DIRECTIONS FOR FUTURE WORK

A future direction of this work lies in further developing the model of electricity distribution networks using MODAM.

- It entails continuing on growing the model and simulating additional possible future trajectories of consumption. This would require the addition of new asset types, and new behaviours representing their use which can be informed by policies;
- Along the expansion of the model, algorithms for the modelling of load consumption at premises could be improved using different modelling methods (e.g. regression analyses, learning algorithms) and by obtaining

more accurate input data which should now be more easily available with the deployment of smart-meters.

As a consequence of further developing the model, future work could also involve structuring the libraries of assets and behaviours from an ontological perspective.

- Indeed, a growing library of assets and behaviours would become a challenge considering the current way of doing so, as mentioned previously. Creating an ontology that wraps around the agent-based structure of the simulation platform could be a solution to this problem.

Other self-organising socio-technical systems offering more challenging topologies and levels of interaction could be implemented with MODAM

- Currently, MODAM's implementation is for a tree-based network of spatially distributed but fixed assets exchanging continuous flows of energy. It is expected that the future grid, with its decentralised systems supplemented by the main grid, will look rather like small-world or scale-free networks. Evaluating how MODAM would perform in terms of ease of extension of the framework for such topologies as well as performance of simulations would be of interest.

MODAM could also be applied to other domains that have simular structures.

- MODAM's architecture was driven by the characteristics of the domain application, which is representing a tree structure over which electricity flows, driven by consumption mainly at the leaves. Other domains have similar characteristics, such as water networks and gas networks. While some of the details of the functioning of these systems differ, the main structure is similar. For example, in the case of water networks, water flows over the network at different rates, depending on the section, to answer the demand at the points of consumption. This demand varies depending on the location, the weather, the time of the day, etc, and can further be impacted by the use of water tanks which are storage units, analogous to batteries. It differs in the way that there cannot be bi-directional flows of water or that water can stay in the pipes while electricity produced needs to match demand exactly at any time. But, these are characteristics that would be represented within the agents' rules, with the model specifically developed for the water domain.

Another future direction is in having greater automation within the framework, for assessing the simulation outputs as well as the models performance.

- With a large number of available options to building simulations, it can become difficult to interpret the output of the simulations. An automatic way to interpret simulation outputs would be beneficial, at least for a few output variables. This automation could be used to perform comparison of scenarios (pairwise or for a larger number of scenarios) to see how one input change would impact the overall output. Because of the large amount of output data, methods from the domain of big data would need to be investigated and used. For this, a module of queries on the data could be created, building on the modularity capability of the framework. The framework already has modules that can transform the data: one module creates *kml* files (Wernecke, 2008) from the simulation output to visualise them in Google Earth, another one calculates statistics on the data using R (R Development Core Team, 2011) to further interpret the results. Additional ones could be developed and their use automated;
- The evaluation of model performance to establish confidence in the models created as expressed in (Bennett et al., 2013) could be automated. This is a separate issue from the verification and validation of MODAM mentioned in the body of this thesis, but rather concerns data and the ABMS created by combining the modules. Because models are being built in an automated manner by bringing components together, each time a modeller sets up a simulation, a new model is created. Depending on the data being used, and the way the components are brought together, models might not be appropriate or reliable. Model performance which can be of quantitative or qualitative nature should be performed on every model created, and this

process could be facilitated by having an automated mechanism that evaluates a set of metrics. A new module, or set of modules, could be designed that would automate the evaluation of the model using simple criteria, and giving a minimum set of metrics that are satisfactory for model performance. This would only be a help to the model performance evaluation, as every model would need to be assessed individually but this would greatly reduce the task of the modeller.

The MODAM core implementation could also be further developed, especially to speed up simulations.

- A parallel scheduler has been presented in this thesis; it uses a fine-grained shared-memory parallel implementation. Our parallel approach could be combined with distributed methods and tested to see if they would improve the speed of simulations.

In order for MODAM to be used more widely by planners, it would be beneficial to better integrate it with planning tools from the distribution network providers.

- For this, key questions that planners typically have could be identified and implemented within a GUI, as a set of pre-defined models so that minimal simulation set up would be required from the decision-makers;
- Also, key indicators that are output of simulations could be identified, in addition to the currently modelled load, voltage and current;
- Finally, metrics for the automation of model performance could be identified by the users which would lead to greater confidence in the use of MODAM by the planners.
- This could be done as a workshop where interested parties could trial the software and provide their input.

Bibliography

Abras, S., Kieny, C., Ploix, S., & Wurtz, F. (2013, 2013). *MAS Architecture for Energy Management: Developing Smart Networks with JADE Platform.* Paper presented at the 2013 IEEE International Conference onSmart Instrumentation, Measurement and Applications (ICSIMA), Kuala Lumpur.

AECOM Australia Pty Ltd. (2012). Impact of Electric Vehicle and Natural Gas Vehicles on the Energy Markets.

AEMC. (2012). Energy Market Arrangements for Electric and Natural Gas Vehicles. Sydney South: Australian Energy Market Commission.

AEMO. (2012). Rooftop PV Information Paper - National Electricity Forecasting (pp. 60): AEMO - Australian Energy Market Operator.

Agile Alliance. (2011). Unit Testing. *Guide to Agile Practices.* Retrieved 05/10/2014, 2014, from http://guide.agilealliance.org/guide/unittest.html

Alam, M. J. E., Muttaqi, K. M., & Sutanto, D. (2013). Mitigation of Rooftop Solar PV Impacts and Evening Peak Support by Managing Available Capacity of Distributed Energy Storage Systems. *IEEE Transactions on Power Systems, 28*(4), 3874-3884. doi: 10.1109/tpwrs.2013.2259269

Albert, R., Albert, I., & Nakarado, G. L. (2004). Structural vulnerability of the North American power grid. *Physical review. E, Statistical, nonlinear, and soft matter physics, 69*(2 Pt 2), 025103. doi: 10.1103/PhysRevE.69.025103

Arefi, A., & Ledwich, G. (2013). *Maximum loadability achievement in SWER networks using optimal sizing and locating of batteries*.

Argonne. Electricity Market Complex Adaptive System (EMCAS). Retrieved 14/03/2011, 2011, from http://www.dis.anl.gov/projects/emcas.html

Argonne. Electricity Market Complex Adaptive System (EMCAS) - Model Introduction. *February 2008 EMCAS Specifications.* Retrieved 14/03/2011, 2011, from http://www.dis.anl.gov/pubs/61084.pdf

Argonne National Laboratory. (2011). Repast Simphony. Retrieved 25/09/2011, 2011, from http://repast.sourceforge.net/repast_simphony.html

Argonne National Laboratory. (2014). The Repast Suite. Retrieved 11/11/2014, 2014, from http://repast.sourceforge.net/

Australian Bureau of Statistics. (2010). Australian Statistical Geography Standard (ASGS): Volume 1 - Main Structure and Greater Capital City Statistical Areas, July 2011 Retrieved 02/05/2014, 2014, from http://www.abs.gov.au/AUSSTATS/abs@.nsf/Latestproducts/7CAFD05E79EB6F81CA257801000C64CD?opendocument

Australian Energy Market Operator. (2013). National Electricity Forecasting Report For the National Electricity Market (pp. 1-68): Australian Energy Market Operator.

Australian Energy Market Operator (AEMO). (2010). *An Introduction to Australia's National Electricity Market* Melbourne: Retrieved from http://www.aemo.com.au/corporate/0000-0262.pdf.

Australian Energy Regulator. (2010). What we do in electricity. Retrieved 23/09/2011, 2011, from http://www.aer.gov.au/content/index.phtml/itemId/659171
Australian Government - Clean Energy Regulator. (2012a). The Large-scale Renewable Energy Target (LRET). Retrieved 14/05/2012, 2012, from http://ret.cleanenergyregulator.gov.au/About-the-Schemes/lret
Australian Government - Clean Energy Regulator. (2012b). The Small-scale Renewable Energy Scheme (SRES). Retrieved 02/04/2012, 2012, from http://ret.cleanenergyregulator.gov.au/About-the-Schemes/Small-scale-Renewable-Energy-Scheme--SRES-/about-sres
Australian Government. (2010). Kyoto Protocol. Retrieved 20/08/2011, 2011, from http://www.climatechange.gov.au/en/government/initiatives/kyoto.aspx
Australian Government. (2011a). National targets. Retrieved 20/08/2011, 2011, from http://www.climatechange.gov.au/government/reduce/national-targets.aspx
Australian Government. (2011b). Renewable Energy Target. Retrieved 23/08/2011, 2011, from http://www.climatechange.gov.au/government/initiatives/renewable-target.aspx
Australian PV Institute, & Australian Renewable Energy Agency. (2013). Australian PV market since April 2001. Retrieved 08/01/2014, 2014, from http://pv-map.apvi.org.au/analyses
Bae, J. W., Lee, G., & Moon, I.-C. (2012, 2012). *Formal specification supporting incremental and flexible agent-based modeling*. Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.
Balci, O. (2013). Introduction to Modeling and Simulation. *ACM Special Interest Group on Simulation and Modeling (SIGSIM) Modeling and Simulation Knowledge Repository (MSKR)*. http://www.acm-sigsim-mskr.org/Courseware/Balci/introToMS.htm
Balci, O., Arthur, J. D., & Ormsby, W. F. (2011). Achieving reusability and composability with a simulation conceptual model. *Journal of Simulation, 5*(3), 157-165. doi: 10.1057/jos.2011.7
Baldwin, C. Y., & Clark, K. B. (2000). *Design Rules: Vol. 1: The Power of Modularity*. Cambridge: MIT Press.
Banks, J., & Chwif, L. (2011). Warnings about simulation. *Journal of Simulation, 5*(4), 279-291. doi: 10.1057/jos.2010.24
Batten, D. F., & Grozev, G. (2006). NEMSIM: Finding Ways to Reduce Greenhouse Gas Emissions Using Multi-Agent Electricity Modelling *Complex science for a complex world: exploring human ecosystems with agents* (pp. 227-252). Canberra: ANU E Press.
Beck, K., & Andres, C. (2005). *Extreme programming explained: embrace change*. Boston, MA: Addison-Wesley.
Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved 01/03/2014, 2014, from http://agilemanifesto.org/
Bedau, M. A. (1997). Weak Emergence. In J. E. Tomberlin (Ed.), *Philosophical Perspectives: Mind, Causation and World* (Vol. 11, pp. 375-399): Malden, MA: Blackwell.

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2003). JADE - A White Paper. *exp in search of innovation, 3*(3).

Bellifemine, F., Caire, G., Trucco, T., & Rimassa, G. (2010). JADE Programmer's Guide. Boston, MA , USA.

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Hoboken, New Jersey, USA: Wiley-Blackwell.

Benali, H., & Ben Saoud, N. B. (2011). Towards a component-based framework for interoperability and composability in Modeling and Simulation. *Simulation, 87*(1-2), 133-148. doi: 10.1177/0037549710373910

Bennett, N. D., Perrin, C., Pierce, S. A., Robson, B., Seppelt, R., Voinov, A. A., . . . Norton, J. P. (2013). Characterising performance of environmental models. *Environmental Modelling & Software, 40* (2013), 1-20. doi: 10.1016/j.envsoft.2012.09.011

Berryman, M. (2008). *Review of Software Platforms for Agent Based Models*. (DSTO-GD-0532). Edinburgh, South Australia, Australia: Defence Science Technology Organisation.

Blewitt, A. (2007). Getting started with Eclipse plug-ins: creating extension points. Retrieved 13/02/213, 2013, from http://www.eclipsezone.com/eclipse/forums/t97608.rhtml

Boait, P. J., Ardestani, B. M., Mark Rylatt, R., & Richard Snape, J. (2013). Managing complexity in the smart grid through a new approach to demand response. *Emergence: Complexity and Organization, 15*(2), 23-37.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D. U. (2006). Complex networks: Structure and dynamics. *Physics Reports, 424*(4), 175-308. doi: 10.1016/j.physrep.2005.10.009

Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America, 99 Suppl 3*(3), 7280-7287. doi: 10.1073/pnas.082080899

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial isystems*. New York: Oxford University Press.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide*. Upper Saddle River, NJ: Addison-Wesley.

Boulaire, F., Utting, M., & Drogemuller, R. (2013a, 18-26 May 2013). *MODAM: A MODular Agent-based Modelling Framework.* Paper presented at the 2nd International Workshop on Software Engineering Challenges for the Smart Grid as part of the 35th International Conference on Software Engineering (ICSE 2013), San Fransisco, CA, USA.

Boulaire, F., Utting, M., & Drogemuller, R. (2013b, 26/08/2013). *Parallel ABM for electricity distribution grids: a case study.* Paper presented at the 1st Workshop on Parallel and Distributed Agent-Based Simulations, Euro-Par 2013, Aachen, Germany.

Boulaire, F., Utting, M., & Drogemuller, R. (2014). Assessment of the impact of technological uptake on an Australian electricity distribution network through simulations using MODAM. *Environmental Modelling & Software*.

Boulaire, F., Utting, M., & Drogemuller, R. (2015a). Dynamic agent composition for large-scale agent-based models. *Complex Adaptive Systems Modeling, 3*(1), 1.

Boulaire, F., Utting, M., & Drogemuller, R. (2015b). Dynamic Agent Composition for Large-Scale Agent-based Models. *Complex Adaptive Systems Modeling*. doi: 10.1186/s40294-015-0007-2

Boulaire, F., Utting, M., Drogemuller, R., Abeygunawardana, A., Ledwich, G., & Bell, J. (2012, 6-7 December 2012). *Planning for the Impact of Distributed Solar Energy on the Grid.* Paper presented at the 50th Annual Conference, Australian Solar Energy Society (Australian Solar Council), Swinburne University of Technology, Melbourne.

Boulaire, F., Utting, M., Drogemuller, R., Ledwich, G., & Ziari, I. (2012, 9-12 December 2012). *A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network.* Paper presented at the 2012 Winter Simulation Conference, Berlin, Germany.

BREE 2013. (2013). *2013 Australian Energy Statistics Data*. Retrieved from: http://www.bree.gov.au/publications/australian-energy-statistics/2013-australian-energy-statistics-data

Briot, J.-P., & Meurisse, T. (2006). A Component-based Model of Agent Behaviors for Multi-Agent-Based Simulations. In P. Stone & G. Weiss (Eds.), *Proceedings of the 7th International Workshop on Multi-Agent-Based Simulation (MABS'06), 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'2006)* (pp. 183–190). New York, NY, USA: Association for Computing Machinery (ACM).

Bunn, D. W., & Oliveira, F. S. (2007). Agent-based analysis of technological diversification and specialization in electricity markets. *European Journal of Operational Research, 181*(3), 1265-1278. doi: 10.1016/j.ejor.2005.11.056

Burger, A. (2014). Germany Sets Three National Solar Records in Two Weeks. *2014*(06/07/2014). Retrieved from Triple Pundit - people, planet, profit

website: http://www.triplepundit.com/2014/06/germany-sets-three-national-solar-records-two-weeks/

Buss, A. (2000). *Component-Based Simulation Modeling.* Paper presented at the 2000 Winter Simulation Conference.

Cai, C., Jahangiri, P., Thomas, A. G., Zhao, H., Aliprantis, D. C., & Tesfatsion, L. (2011, 24-29/07/2011). *Agent-Based Simulation of Distribution Systems with High Penetration of Photovoltaic Generation*, San Diego, CA.

Camacho, J., Guimerà, R., & Nunes Amaral, L. A. (2002). Robust patterns in food web structure. *Physical review letters, 88*(22), 228102. doi: 10.1103/PhysRevLett.88.228102

Carneiro, D., Novais, P., Costa, R., & Neves, J. (2010). Developing intelligent environments with OSGi and JADE. In M. Bramer (Ed.), *Artificial Intelligence in Theory and Practice III* (Vol. 331, pp. 174-183): Springer Berlin Heidelberg.

Casey, S. D. (2011). How to Determine the Effectiveness of Hyper-Threading Technology with an Application. *Intel® Developer Zone.* Retrieved 26/07/2013, 2013, from http://software.intel.com/en-us/articles/how-to-determine-the-effectiveness-of-hyper-threading-technology-with-an-application/

Castiglione, F. (2006). Agent based modeling. *Scholarpedia, 1*(10), 1562. doi: doi::10.4249/scholarpedia.1562

Chan, W. K. V., Young-Jun, S., & Macal, C. M. (2010, 2010). *Agent-based simulation tutorial - simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation*. Paper presented at the 2010 Winter Simulation Conference, Baltimore, MD.

Chappin, E. J. L., & Dijkema, G. P. J. (2010). Agent-based modelling of energy infrastructure transitions. *International Journal of Critical Infrastructures, 6*(2), 106-130. doi: 10.1504/ijcis.2010.031070

Chassin, D. P., Fuller, J. C., & Djilali, N. (2014). GridLAB-D: An agent-based simulation framework for smart grids. *Applied Mathematics*(Journal Article).

Chassin, D. P., Schneider, K., & Gerkensmeyer, C. (2008). *GridLAB-D: An open-source power systems modeling and simulation environment*. Paper presented at the Transmission and Distribution Conference and Exposition.

Clean Energy Regulator. (2014a, 05/03/2014). Clean Energy Regulator - Small-scale installations by postcode. Retrieved 25/03/2014, 2014, from http://ret.cleanenergyregulator.gov.au/REC-Registry/Data-reports

Clean Energy Regulator. (2014b). Small-scale installations by postcode. Retrieved 11/04/2014, 2014, from http://ret.cleanenergyregulator.gov.au/REC-Registry/Data-reports

Collier, N. (2013). Repast HPC Manual. Retrieved 01/09/2014, 2014, from http://repast.sourceforge.net/docs/RepastHPCManual.pdf

Collier, N. (2014). Interface ContextBuilder<T>. Retrieved 03/01/2015, 2015, from http://repast.sourceforge.net/docs/api/repast_simphony/index.html

Connolly, D. (2009). COMPOSE. Retrieved 08/05/2012, 2012, from http://www.dconnolly.net/research/planning/tools/compose.html

Connolly, D., Lund, H., Mathiesen, B. V., & Leahy, M. (2010). A review of computer tools for analysing the integration of renewable energy into various energy systems. *Applied Energy, 87*(4), 1059-1082. doi: 10.1016/j.apenergy.2009.09.026

Conway, J. (1970). The Game of Life. *Scientific American, 223*(4), 120-123.

Conzelmann, G., Boyd, G., Koritarov, V., & Veselka, T. (2005). Multi-Agent Power Market Simulation using EMCAS. *IEEE power engineering society general meeting, 3*, 2829-2834.

Cordasco, G., Chiara, R. D., Raia, F., Scarano, V., Spagnuolo, C., & Vicidomini, L. (2013). *Designing computational steering facilities for distributed agent based simulations*. Paper presented at the Proceedings of the 2013 ACM SIGSIM conference on Principles of advanced discrete simulation, Montreal, Quebec, Canada.

CSIRO Future Grid Forum. (2013). Change and choice: CSIRO.

Cuevas-Cubria, C., Schultz, A., Petchey, R., Maliyasena, A., & Sandu, S. (2010). *Energy in Australia 2010*. (ISSN 1833-038). Australian Government - Department of Resources, Energy and Tourism.

Czerwonka, J. (2014). Pairwise Testing. Retrieved 04/05/2014, 2014, from http://www.pairwise.org/tools.asp

del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G. (2008). Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *Ieee Transactions On Evolutionary Computation, 12*(2), 171-195. doi: 10.1109/tevc.2007.896686

Dijkstra, E. W. (1982). EWD 447: On the role of scientific thought. *Selected Writings on Computing: A Personal Perspective*, 60-66. doi: citeulike-article-id:2490230

Dingsøyr, T., Dybå, T., & Moe, N. (2010). Agile Software Development: An Introduction and Overview. In T. Dingsøyr, T. Dybå & N. B. Moe (Eds.), *Agile Software Development* (pp. 1-13). Berlin, Germany: Springer Berlin Heidelberg.

Douthitt, R. A. (1989). An Economic Analysis Of The Demand For Residential Space Heating Fuel In Canada. *Energy, 14*(4), 187-197. doi: 10.1016/0360-5442(89)90062-5

Dow, L., Marshall, M., Le, X., Aguero, J. R., & Willis, H. L. (2010, 25-29 July 2010). *A novel approach for evaluating the impact of electric vehicles on the power distribution system.* Paper presented at the Power and Energy Society General Meeting, 2010 IEEE.

Duffie, J. A., & Beckman, W. A. (2006). *Solar engineering of thermal processes*. Hoboken, N.J: Wiley.

Elliott, D. (2010). Sustainable Energy Systems: Linking the Local to the Global. *Emergence : Complexity and Organization*, *12*(2), 7.

EMD International A/S. (2012). energyPRO. Retrieved 08/05/2012, 2012, from http://www.emd.dk/energyPRO/Frontpage

Endecon Engineering, & Regional Economic Research Inc. (2001). A Guide to Photovoltaic (PV) System Design and Installation (pp. 1-39): California Energy Commission.

EnergyPLAN. (2012). EnergyPLAN - Advanced Energy System Analysis Computer Model. Retrieved 03/05/2012, 2012

Ergon Energy. (2010). Network Management Plan Part B: Electricity Supply for Regional Queensland 2010-11 to 2014-15.

Ergon Energy. (2011). Information Guide for Standard Control Services Prices - 1 July 2011 to 30 June 2012. Brisbane: Ergon Energy Corporation Limited.

Ergon Energy. (2013a). Corporate profile. Retrieved 02/06/2013, 2013, from https://www.ergon.com.au/about-us/who-we-are/our-company/corporate-profile

Ergon Energy. (2013b). Distribution Annual Planning Report 2013/14 to 2017/18 (Vol. Part B, pp. 1-335). Brisbane: Ergon Energy Corporation Limited.

Fan, S., & Hyndman, R. J. (2013). Forecasting long-term peak half-hourly electricity demand for Queensland. Melbourne, Australia: Monash University.

Ferber, J. (1999). *Multi-agent Systems: an Introduction to Distributed Artificial Intelligence*.

Foley, A. M., Ó Gallachóir, B. P., Hur, J., Baldick, R., & McKeogh, E. J. (2010). A strategic review of electricity systems models. *Energy*, *35*(12), 4522-4530. doi: 10.1016/j.energy.2010.03.057

Fowler, M. (2004, 23/01/2004). Inversion of Control Containers and the Dependency Injection pattern. Retrieved 05/12/2013, 2013, from http://martinfowler.com/articles/injection.html

Gamma, E. (2009). *Design patterns: elements of reusable object-oriented software*. Boston: Addison-Wesley.

GAMS Development Corporation. (2012). GAMS. Retrieved 05/10/2012, 2012, from http://www.gams.com/

Genoese, M., Sensfuß, F., Weidlich, A., Möst, D., & Rentz, O. (2005, September 7-9, 2005). *Development of an agent-based model to analyse the effect of renewable energy on electricity markets*. Paper presented at the 19. International Conference Informatics for Environmental Protection, EnviroInfo, Brno, Czech Republik.

Gnansounou, E., Pierre, S., Quintero, A., Dong, J., & Lahlou, A. (2007). A Multi-Agent Approach for Planning Activities in Decentralized Electricity Markets. *Knowledge-Based Systems, 20*(4), 406-418. doi: 10.1016/j.knosys.2006.06.004

Gomez, J. F., Khodr, H. M., De Oliveira, P. M., Ocque, L., Yusta, J. M., Villasana, R., & Urdaneta, A. J. (2004). *Ant colony system algorithm for the planning of primary distribution circuits*.

Gong, Z. Y., Tang, W. W., Bennett, D. A., & Thill, J. C. (2013). Parallel agent-based simulation of individual-level spatial interactions within a multicore computing environment. *INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE, 27*(6), 1152-1170. doi: 10.1080/13658816.2012.741240

Grimm, V., Huse, G., Huth, A., Jepsen, J. U., Jørgensen, C., Mooij, W. M., . . . Heinz, S. K. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling, 198*(1), 115-126. doi: 10.1016/j.ecolmodel.2006.04.023

Grohnheit, P. E. (1999). *Energy policy responses to the climate change challenge: The consistency of European CHP, renewables and energy efficiency policies (also published as Risø-R-1147(EN))*. Luxembourg: Office for Official Publications of the European Communities.

H2RES MODEL. (2009). H2RES Model. Retrieved 03/05/2012, 2012, from http://powerlab.fsb.hr/h2res/

Haas, A., & White, F. (2012). GridLAB-D: A One-of-a-Kind Energy Grid Simulator. *Transmission & Distribution World*. Retrieved from http://qut.summon.serialssolutions.com/2.0.0/link/0/eLvHCXMwVZ3BCsIwDIaLIHjxMtBefYGOduva7SwOQfA0cR6bpPU2EPf-mE0FvSWHlkLK_5E2IULsGJo-Wm2BKiLSTaTSJKDpE8Zx2ox_j20_at5mYhGHjejaQ7c_qs8wAHVn6ClwiIaCRQYeb1szWR3bQScq6uidT5igALRFAkDGVuIV3lXec-qIJpitWIepZnwY594ykmKZOMBRTqIr-QBSrPrmeqlvp_Pbzb5u_pwboPLHKFnj5_uhyly_AM4POoc

Hall, P. (2011, 24 September 2011). Queensland State Government admits electricity grid failing to cope with solar power systems. *Courier Mail*. Retrieved from http://www.couriermail.com.au/news/queensland/solar-dream-caught-in-gridlock/story-e6freoof-1226144903889

Hamill, L. (2010). Agent-based modelling: The next 15 years. *Journal of Artificial Societies and Social Simulation, 13*(4), 7.

Hayes, I., Flinn, B., Gimson, R., King, S., Morgan, C., Sorensen, I. H., & Sufrin, B. (1986). *Specification Case Studies* (I. Hayes Ed. 2nd ed.): Pearson Education Limited.

Helsinger, A., Wright, T., & Ieee. (2005). Cougaar: A robust configurable multi agent platform *2005 IEEE Aerospace Conference, Vols 1-4* (pp. 3129-3138).

Higgins, A., Paevere, P., Gardner, J., & Quezada, G. (2012). Combining choice modelling and multi-criteria analysis for technology diffusion: An application to the uptake of electric vehicles. *Technological Forecasting and Social Change, 79*(8), 1399-1412. doi: http://dx.doi.org/10.1016/j.techfore.2012.04.008

HOMER Energy. (2012). Energy Modeling Software for Hybrid Renewable Energy Systems. Retrieved 03/05/2012, 2012, from http://www.homerenergy.com/

Hopkinson, K., Xiaoru, W., Giovanini, R., Thorp, J., Birman, K., & Coury, D. EPOCHS: a Platform for Agent-Based Electric Power and Communication Simulation Built from Commercial Off-the-shelf Components. *21*, 548-558. doi: 10.1109/tpwrs.2006.873129

House-Peters, L. A., & Chang, H. (2011). Urban Water Demand Modeling: Review of Concepts, Methods, and Organizing Principles. *Water Resources Research, 47*(5), W05401. doi: 10.1029/2010wr009624

Institute for Energy and Transport. (2014, 18/07/2014). Agent Based Modelling for Smart Grids. Retrieved 20/07/2014, 2014, from http://ses.jrc.ec.europa.eu/agent-based-modelling-smart-grids

Intelligent Energy Systems. (2011). Prophet Version 2011. In I. E. Systems (Ed.). Crows Nest, NSW, Australia.

International Energy Agency. (2011). Technology Roadmaps - Smart Grids. 52.

International Energy Agency (Producer). (2014, 07/04/2014). Energy Technology Perspectives 2012 Data Visualisation. [Data Visualisation - Interactive Graphs] Retrieved from http://www.iea.org/etp/explore/

Jun, Z., Junfeng, L., Jie, W., & Ngan, H. W. (2011). A multi-agent solution to energy management in hybrid renewable energy generation system. *Renewable Energy, 36*(5), 1352-1363. doi: 10.1016/j.renene.2010.11.032

Kelly, R. A., Rizzoli, A. E., Delden, H., Voinov, A. A., Jakeman, A. J., Barreteau, O., . . . Maier, H. R. (2013). Selecting among five common modelling approaches for integrated environmental assessment and management. *Environmental Modelling & Software, 47*(Journal Article), 159-181. doi: 10.1016/j.envsoft.2013.05.005

Kirby, B., & Milligan, M. (2008). Facilitating Wind Development: The Importance of Electric Industry Structure. Golden, Colorado: National Renewable Energy Laboratory.

Klügl, F., & Bazzan, A. L. C. (2012). Agent-based modeling and simulation. *AI Magazine, 33*(3), 29-40.

Komor, P. (2009). Wind and Solar Electricity: Challenges and Opportunities. In PEW Center on Global Climate Change (Ed.). Boulder: University of Colorado.

Ledwich, G., Drogemuller, R., Utting, M., Ziari, I., Boulaire, F., & Abeygunawardana, A. (2012). Planning Future Energy Grids : Renewables - Project Milestone Report (P. Engineering, Trans.) (pp. 1-36). Brisbane, Australia: Queensland University of Technology.

Lilley, B., Szatow, A., & Jones, T. (2009). Intelligent grid - a value proposition for distributed energy in Australia (N. R. F.-E. Transformed, Trans.). In CSIRO (Ed.), *National Research Flagships - Energy Transformed*: CSIRO.

Linkola, L., Andrews, C. J., & Schuetze, T. (2013). An Agent Based Model of Household Water Use. *WATER, 5*(3), 1082-1100. doi: 10.3390/w5031082

Liu, B. Y. H., & Jordan, R. C. (1960). The interrelationship and characteristic distribution of direct, diffuse and total solar radiation. *Solar Energy, 4*(3), 1-19. doi: 10.1016/0038-092x(60)90062-1

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). MASON: A Multi-Agent Simulation Environment. *Simulation: Transactions of the society for Modeling and Simulation International, 82*(7), 517-527.

Lund, H., & Mathiesen, B. V. (2009). Energy system analysis of 100% renewable energy systems—The case of Denmark in years 2030 and 2050. *Energy, 34*(5), 524-531. doi: 10.1016/j.energy.2008.04.003

Ma, T., & Nakamori, Y. (2009). Modeling technological change in energy systems – From optimization to agent-based modeling. *Energy, 34*(7), 873-879. doi: 10.1016/j.energy.2009.03.005

Macal, C. M., & North, M. J. (2005, 2005). *Agent-Based Modeling And Simulation.* Paper presented at the 2005 Winter Simulation Conference.

Macal, C. M., & North, M. J. (2006, December 3-6, 2006). *Tutorial On Agent-Based Modeling And Simulation Part 2: How To Model With Agents.* Paper presented at the Winter Simulation Conference, Monterey, California, USA.

Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*(4), 151-162.

Manyika, J., Chui, M., Bughin, J., Dobbs, R., Bisson, P., & Marrs, A. (2013). Disruptive technologies: Advances that will transform life, business, and the global economy (pp. 1-30): McKinsey Global Institute.

Maria, A. (1997, 1997). *Introduction to Modeling and Simulation.* Paper presented at the Winter Simulation Conference, WSC 97, Atlanta, GA, USA.

Marshall, A. C., Boffey, T. B., Green, J. R., & Hague, H. (1991). Optimal design of electricity distribution networks. *IEE Proceedings C Generation, Transmission and Distribution, 138*(1), 69. doi: 10.1049/ip-c.1991.0009

McAffer, J., & Lemieux, J.-M. (2006). *Eclipse Rich Client Platform: designing, coding, and packaging Java applications*. Upper Saddle River, NJ: Addison-Wesley.

McArdle, M. (2013). *Future power network gets a little help from "GUSS"*. Brisbane: Retrieved from http://statements.qld.gov.au/Statement/2013/10/3/future-power-network-gets-a-little-help-from-guss.

Meurisse, T., & Peschanski, F. (2007). Architectural Design of Component-Based Agents: A Behavior-Based Approach *Programming Multi-Agent Systems* (Vol. 4411, pp. 71-90). Berlin, Heidelberg: Springer Berlin Heidelberg.

Mihalakakou, G., Santamouris, M., & Tsangrassoulis, A. (2002). On the energy consumption in residential buildings. *Energy and Buildings, 34*(7), 727-736. doi: 10.1016/s0378-7788(01)00137-2

Moglia, M., Perez, P., & Burn, S. (2010). Modelling an Urban Water System on the Edge of Chaos. *Environmental Modelling & Software, 25*(12), 1528-1538. doi: 10.1016/j.envsoft.2010.05.002

Morton, A. (2003, 27-30/09). *A fast 'do-it-yourself' load flow algorithm for power systems with sparse topology*. Paper presented at the AUPEC 2003

Australasian Universities Power Engineering Conference, Christchurch, New Zealand.
Najlis, R., Janssen, M. A., & Parker, D. C. (2001, 04-07/10/2001). *Software Tools and Communication Issues.* Paper presented at the Proceedings of a Special Workshop on Land-Use/Land-Cover Change, Irvine, California.
National Renewable Energy Laboratory. (2012). PVWatts - A Performance Calculator for Grid-Connected PV Systems. Retrieved 05/01/2012, 2012, from http://rredc.nrel.gov/solar/calculators/PVWATTS/version1/version1_index.html
National Renewable Energy Laboratory. (2014). PVWatts - How to Change Parameters in Legacy Calculators. Retrieved 25/03/2014, 2014, from http://www.nrel.gov/rredc/pvwatts/changing_parameters.html
Ndiaye, D., & Gabriel, K. (2011). Principal component analysis of the electricity consumption in residential dwellings. *Energy and Buildings, 43*(2-3), 446-453. doi: 10.1016/j.enbuild.2010.10.008
Nicolis, G., & Rouvas-Nicolis, C. (2007). Complex systems. *Scholarpedia, 2*, 1473.
Nikolai, C., & Madey, G. (2009). Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation, 12*(2), 2.
Nikolic, I., & Dijkema, G. P. J. (2010). On the development of agent-based models for infrastructure evolution. *International Journal of Critical Infrastructures, 6*(2), 148-148. doi: 10.1504/ijcis.2010.031072
North, M., Conzelmann, G., Koritarov, V., Macal, C., Thimmapuram, P., & Veselka, T. (2002). *E-laboratories : agent-based modeling of electricity markets*. Paper presented at the American Power Conference, Chicago, IL (US).
North, M. J. (2013). A theoretical formalism for analyzing agent-based models. *Complex Adaptive Systems Modeling, 2*(1), 3-3. doi: 10.1186/2194-3206-2-3
North, M. J., & Macal, C. M. (2007). *Managing Business Complexity*. New York, NY: Oxford University Press.
Oncor. (2012). Generating Electricity: Geothermal - Pathway of Power. Retrieved 14/06/2012, 2012, from http://www.oncor.com/community/knowledgecollege/energy_library/pathway/default.aspx
Ören, T. (2011). A Critical Review of Definitions and About 400 Types of Modeling and Simulation. *SCS M&S Magazine, 3*, 142-151.
OSGi Alliance. (2013). OSGi Alliance. Retrieved 01/02/2013, 2013, from http://www.osgi.org/Main/HomePage
Paevere, P., Higgins, A., Ren, Z., Horn, M., Grozev, G., & McNamara, C. (2014). Spatio-temporal modelling of electric vehicle charging demand and impacts on peak household electrical load. *Sustainability Science, 9*(1), 61-76. doi: 10.1007/s11625-013-0235-3
Page, E., & Opper, J. (1999). *Observations on the complexity of composable simulation.* Paper presented at the Winter Simulation Conference Proceedings, 1999, Phoenix, AZ.

Parker, J. (2007, 2007). *A flexible, large-scale, distributed agent based epidemic model*. Paper presented at the 2007 Winter Simulation Conference, Washington, DC, USA.

Parkinson, G. (2013). Interview: Vector CEO Simon Mackenzie. *REneweconomy*. Retrieved 05/10/2013, 2013, from http://reneweconomy.com.au/2013/interview-vector-ceo-simon-mackenzie-69896

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM, 15*(12), 1053-1058. doi: 10.1145/361598.361623

Parnas, D. L., & Clements, P. C. (1986). A Rational Design Process: How and Why to Fake It. *Software Engineering, IEEE Transactions on, SE-12*(2), 251-257. doi: 10.1109/tse.1986.6312940

Parry, H. R. (2012). Agent Based Modeling, Large Scale Simulations (pp. 76-87). New York, NY: Springer New York.

Petty, M. D., & Weisel, E. W. (2003). *A Composability Lexicon*. Paper presented at the 2003 Spring Simulation Interoperability Workshop, Kissimmee, Florida. http://www.sisostds.org/DigitalLibrary.aspx?EntryId=25018

Pezeshki, H., Wolfs, P., & Johnson, M. (2011, 13-16 Nov. 2011). *Multi-Agent Systems for Modeling High Penetration Photovoltaic System Impacts in Distribution Networks*. Paper presented at the Innovative Smart Grid Technologies Asia (ISGT), 2011 IEEE PES.

Powertech Labs Inc. (2012). DSATools - Dynamic Security Assessment Software. Retrieved 23/05/2012, 2012, from http://www.dsatools.com/

Quarantotto, E., & Caire, G. (2010). JADE OSGi Guide.

Queensland Government - Office of Clean Energy. (2011). Solar Bonus Scheme. Retrieved 02/04/2012, 2012, from http://ret.cleanenergyregulator.gov.au/About-the-Schemes/Small-scale-Renewable-Energy-Scheme--SRES-/about-sres

Queensland Government. (2009). Securing Queensland's Energy Future: Regulation for Electricity Demand Management. In Queensland Government - Office of Climate Change (Ed.).

Queensland Government. (2011). *Queensland Government Populations Projectiosn to 2056: Queensland and Statistical Divisions*. Retrieved from http://www.qgso.qld.gov.au/products/publications/qld-govt-pop-proj-qld-sd/qld-govt-pop-proj-2056-qld-sd-2011.pdf.

R Development Core Team. (2011). R: A Language and Environment for Statistical Computing. Retrieved from http://www.R-project.org

Railsback, S. F., Lytinen, S. L., & Jackson, S. K. A template model for ABM platforms. Retrieved 28/10/2013, 2013, from http://condor.depaul.edu/slytinen/abm/StupidModel/

Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. *Simulation, 82*(9), 609-623.

Rath-Nagel, S., & Stocks, K. (1982). Energy modelling for technology assessment: the MARKAL approach. *Omega, 10*(5), 493-505. doi: 10.1016/0305-0483(82)90006-8

Ravn, H. (2012). Balmorel Energy System Model. Retrieved 08/05/2012, 2012, from http://www.eabalmorel.dk/

Ren, F., Zhang, M., & Sutanto, D. (2013). A Multi-Agent Solution to Distribution System Management by Considering Distributed Generators. *IEEE Transactions on Power Systems, 28*(2), 1442-1451. doi: 10.1109/tpwrs.2012.2223490

Riedy, C., & Partridge, E. (2006). Study of Factors Influencing Electricity Use in Newington (pp. 1-114). Sydney: Institute for Sustainable Futures, UTS.

Rinaldi, S. M., Peerenboom, J. P., & Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine, 21*, 11.

ROAM Consulting. (2011). Projections of Electricity Generation in Australia to 2050. Brisbane: ROAM Consulting - Energy Modelling Expertise.

ROAM Consulting. (2012). 2-4-C Lite. Toowong, Qld, Australia: ROAM Consulting. Retrieved from http://www.roamconsulting.com.au/24CLite/index24C.php

Royce, W. W. (1987). *Managing the development of large software systems: concepts and techniques*. Paper presented at the Proceedings of the 9th international conference on Software Engineering, Monterey, California, USA.

Saddler, H. (2013). Power Down - Why is electricity consumption decreasing? (Vol. 14, pp. 1-73): The Australia Institute.

Sarvapali, D. R., Perukrishnen, V., Alex, R., & Nicholas, R. J. (2012). Putting the 'Smarts' into the Smart Grid: A Grand Challenge for Artificial Intelligence. *Association for Computing Machinery. Communications of the ACM, 55*(4), 86.

Schelling, T. C. (1971). Dynamic Models of Segregation *Journal of Mathematical Sociology, 1*, 143-186.

Schutte, S., Scherfke, S., & Sonnenschein, M. (2012, 19-20 April 2012). *mosaik - Smart Grid Simulation API*. Paper presented at the SmartGreens 2012.

Schutte, S., & Sonnenschein, M. (2012, 2012). *Mosaik - Scalable Smart Grid scenario specification*. Paper presented at the Winter Simulation Conference, Berlin, Germany.

Shi, J., Ren, A., & Chen, C. (2009). Agent-Based Evacuation Model of Large Public Buildings under Fire Conditions. *Automation in Construction, 18*(3), 338-347. doi: 10.1016/j.autcon.2008.09.009

Shukla, P. R. (2013). Review of linked modelling of low-carbon development, mitigation and its full costs and benefits *MAPS Reserach Paper*: MAPS.

Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society, 106*(6), 467-482.

Smart Grid Smart City. (2012). Customer Applications Stream: Electric Vehicles *Monitoring and Measurement Report* (Vol. 3).

Solé, R. V., & Montoya, M. (2001). Complexity and fragility in ecological networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences, 268*(1480), 2039-2045. doi: 10.1098/rspb.2001.1767

Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008). *EMF: Eclipse Modeling Framework* Boston, MA, USA: Addison-Wesley Professional.

Swan, L. G., & Ugursal, V. I. (2009). Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. *Renewable & Sustainable Energy Reviews, 13*(8), 1819-1835. doi: 10.1016/j.rser.2008.09.033

Szyperski, C. (1997). *Component software: beyond object-oriented programming*. New York: ACM Press.

Tang, F., & Ren, A. (2008). Agent-Based Evacuation Model Incorporating Fire Scene and Building Geometry. *Tsinghua Science & Technology, 13*(5), 708-714. doi: 10.1016/s1007-0214(08)70112-3

The Eclipse Foundation. (2011). Agent Modeling Platform. Retrieved 25/09/2011, 2011, from http://www.eclipse.org/amp/

The Eclipse Foundation. (2012). About the Eclipse Foundation. Retrieved 27/02/2012, 2012, from http://www.eclipse.org/org/

Thomas Stober, & Hansmann, U. (2010). Overview of Agile Software Development *Agile Software Development: Best Practices for Large Software Development Projects* (pp. 35-39). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tonkoski, R., Turcotte, D., & El-Fouly, T. H. M. (2012). *Impact of High PV Penetration on Voltage Profiles in Residential Neighborhoods*.

Ustun, T. S., Zayegh, A., & Ozansoy, C. (2013). Electric vehicle potential in Australia: Its impact on smartgrids. *IEEE Industrial Electronics Magazine, 7*(4), 15-25. doi: 10.1109/mie.2013.2273947

van Dam, K. H., Nikolic, I., & Lukszo, Z. (2012). *Agent-Based Modelling of Socio-Technical Systems*. Dordrecht: Springer Netherlands.

Vázquez, A., Pastor-Satorras, R., & Vespignani, A. (2002). Large-scale topological and dynamical properties of the Internet. *Physical review. E, Statistical, nonlinear, and soft matter physics, 65*(6 Pt 2), 066130. doi: 10.1103/PhysRevE.65.066130

Veneman, J. G., Oey, M. A., Kortmann, L. J., Brazier, F. M., & De Vries, L. J. (2011, 27-30 June 2011). *A review of agent-based models for forecasting the deployment of distributed generation in energy systems.* Paper presented at the 2011 Grand Challenges on Modeling and Simulation Conference, the Hague, Netherlands

Verdant Vision. (2012). AEMC Review of Energy Market Arrangements for Electric and Natural Gas Vehicles: Verdant Vision comments on the AEMC Issues Paper and AECOM's Initial Advice (pp. 23): Verdant Vision.

Vogel, L. (2012, 08/05/2012). OSGi Modularity - Tutorial. Retrieved 24/09/2012, 2012, from http://www.vogella.com/articles/OSGi/article.html

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. Urbana and London: University of Illinois Press.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature, 393*(6684), 440-442. doi: 10.1038/30918

Weidlich, A. (2008). *Engineering interrelated electricity markets: an agent-based computational approach*. Heidelberg: Springer [distributor].

Wernecke, J. (2008). *The KML handbook; geographic visualization for the Web* (Vol. 1): Addison-Wesley Professional.

Wikipedia. (2011, 9 August 2011). Agent-based model. Retrieved 23/08/2011, 2011, from http://en.wikipedia.org/wiki/Agent-based_model

Wikipedia. (2012a). Complex adaptive system. Retrieved 25/05/2012, 2012, from http://en.wikipedia.org/wiki/Complex_system

Wikipedia. (2012b, 20 May 2012). Electric power distribution. Retrieved 14/06/2012, 2012, from http://en.wikipedia.org/wiki/Electric_power_distribution

World Nuclear Association. (2010). Australia's Electricity. Retrieved 25/08/2011, 2011, from http://www.world-nuclear.org/info/inf64.html

Zeigler, B. P. (1976). *Theory of modelling and simulation*. New York, N.Y: Wiley.

Zhang, Y., Zhang, B., & Bi, J. (2011). An adaptive agent-based modeling approach for analyzing the influence of transaction costs on emissions trading markets. *Environmental Modelling and Software, 26*(4), 482-491. doi: 10.1016/j.envsoft.2010.10.011

Zheng, M., Meinrenken, C. J., & Lackner, K. S. (2014). Agent-based model for electricity consumption and storage to evaluate economic viability of tariff arbitrage for residential sector demand response. *Applied Energy, 126*(Journal Article), 297-306. doi: 10.1016/j.apenergy.2014.04.022

Ziari, I., Ledwich, G., Ghosh, A., Cornforth, D., & Wishart, M. (2010). Optimal allocation and sizing of capacitors to minimize the transmission line loss and to improve the voltage profile. *Computers and Mathematics with Applications, 60*(4), 1003-1013. doi: 10.1016/j.camwa.2010.03.042

Ziari, I., Ledwich, G., Ghosh, A., & Platt, G. (2012). Integrated Distribution Systems Planning to Improve Reliability Under Load Growth. *IEEE Transactions on Power Delivery, 27*(2), 757-765. doi: 10.1109/tpwrd.2011.2176964

Appendices

Appendix A

MODAM: A MODular Agent-Based Modelling Framework

This appendix contains a paper presented at the 2nd International Workshop on Software Engineering Challenges for the Smart Grid (SE4SG), at the IEEE International Conference on Software Engineering (ICSE), San Fransisco.

The MODAM architecture is introduced. This paper was further extended to form two other papers that are in Chapter 5 and 6 of this thesis.

MODAM: A MODular Agent-Based Modelling Framework

Fanny Boulaire, Mark Utting, Robin Drogemuller

Faculty of Creative Industries
Queensland University of Technology
Brisbane, Australia

Abstract**—**Designing the smart grid requires combining varied models. As their number increases, so does the complexity of the software. Having a well thought architecture for the software then becomes crucial. This paper presents MODAM, a framework designed to combine agent-based models in a flexible and extensible manner, using well known software engineering design solutions (OSGI specification [1] and Eclipse plugins [2]). Details on how to build a modular agent-based model for the smart grid are given in this paper, illustrated by an example for a small network.

Index Terms—Modular, Agent-Based Model, OSGi

I. INTRODUCTION

The future grid is going to be smart, where information and communication technology will allow the automation of the delivery of electricity in an efficient, reliable and sustainable manner [3]. To plan such a grid, it is necessary to select the appropriate components that are able to communicate useful information, to understand their most appropriate placement and finally to tune them so that they can be used in their most efficient manner. These three phases have been considered in a large project the work presented in this paper belongs to [4]. Modelling and simulation has been used in this project to represent the network and trial scenarios to understand its possible evolution; combining agent-based modelling and particle swarm optimisation to model the overall distribution grid.

The work presented in this paper concentrates on the specifics of the agent-based model, implemented in a modular manner. Agent-based modelling has been chosen to model the design of the smart grid for its capacity to describe the components at various scales (defining the granularity of the model) and their behaviour (using decision-making heuristics, learning rules or adaptive processes which make them autonomous). The contribution of this paper lies in the fact that this agent-based model is built in a modular manner, taking advantage of good software engineering practices to build extensible and flexible software. That way, models can evolve as the information relating to the smart grid changes or becomes available or more accurate. Also, models can be implemented by different groups for different analysis types and combined to obtain a more complete system representation. For this, the modules need to follow the guidelines described in this paper, implemented in Java. The MODAM (MODular Agent-based Modelling) framework gives the backbone structure of the modular implementation of ABMs, allowing modules to communicate via named data values and user-defined Java interfaces.

In the first part of this paper, the different challenges faced when building a simulation tool for a smart grid are presented, along with the justification for using an agent-based model and building the software in a modular manner. Details on the architecture of the modular agent-based model (MODAM) follow. The different concepts introduced are illustrated using an example for a small network with solar panels.

II. WHY USE A MODULAR AGENT-BASED MODEL TO DESCRIBE THE SMART GRID?

This section places the context of the work presented here, followed by the justification in using agent-based modelling to represent the smart grid. It then describes the solution in implementing it using a modular approach.

A. Context of the Project and Challenges from Studying such a Grid

The scope of the project is to understand and model the future electrical distribution network managed by Ergon Energy, one of the two electricity distribution companies that serve the state of Queensland, Australia. Ergon's distribution network covers 1.7 million square kilometres and provides power to approximately 650,000 homes and businesses across regional and rural Queensland. The network consists of approximately 150,000 kilometres of power lines and associated distribution infrastructure. Ergon also owns and operates 33 stand-alone power stations that power isolated communities across Queensland that are not connected to the main electricity grid. Two types of network are used: 3 phase network as well as SWER (Single Wired Earth Return). The consumers are of different types (residential, commercial, and industrial) and can also be producers via the use of decentralised generators (solar panels, batteries…).

Having described this system as such, it becomes clear that modelling such a grid poses challenges in terms of modelling techniques and software implementation constraints. Table 1 highlights a few of the challenges when modelling the smart grid, and how they translate in terms of the modelling and software implementation requirements (data and modelling).

Table 1 – Some challenges of the smart grid in the context of our project, and their translation in terms of modelling requirements.

Challenges of the Smart Grid	Technical constraint/Geographical scale
	• Different technical systems
	    o 3 phase systems vs. SWER networks
	• Distance constraints between components –network density variability
	• Variation in terms of usage and load types
	Behaviour of the system
	• From centralised to decentralised
	    o Producers and Consumers at same location
	    o Bi-directional flow of information
	• Changes in terms of usage behaviour
	    o Incentives for usage at different times from time of use tariffs
	    o Changes in net usage behaviour from decentralised generators (PV, battery…)
	Technical - feedback loops
	• Higher response from the controllers to changes in consumptions

	Feedback loop and autonomous response

Data and Modelling requirements	Data requirements
	• Different types of information
	    o Different data for the network topology (SWER, 3 phase)
	    o qualitative vs. quantitative information
	    o Different time scales (records every 1/2 hour, 5 minutes…)
	• Different databases holding the information
	• Large datasets to manage and analyse
	
	Modelling/Analysis requirements
	• Need a good representation of the system actors
	    o Capture individual behaviours
	    o Capture interactions of behaviours
	• Large variation – no averages

B. A Technical Solution to Implementing a Smart-Grid Simulation Tool

From this list of data and modelling requirements, a modular agent-based model was chosen as the solution to represent the smart grid. Reasons for such a choice are given below.

1) The Use of Agent-Based Modelling to Model the Smart-Grid: According to House-Peters [5], the popularity of ABMs for the analysis of complex systems is mainly due to their ability: “to incorporate both spatially and temporally explicit data, to model bidirectional relations between individual human agents and the macrobehavior of the social or environmental system being modeled, to capture emerging patterns at higher scales of the system that result from interactions at lower levels, and to blend qualitative and quantitative approaches”. These are some of the reasons agent-based modelling was chosen for modelling the smart grid, as they are answering some of the challenges described in Table 1 (varying types of data - spatial and time, individual behaviours and interactions, qualitative and quantitative information).

Other considerations for choosing agent-based modelling were that agent-based modelling can model individual components and their individual behaviour which other modelling techniques cannot. For example, statistical techniques which are based on analysing large samples of individuals of similar behaviour cannot capture appropriately networks that are sparsely populated, and for which electricity usage can differ greatly. Indeed, SWER networks often have 30km between each residence with only 20 or 30 customers under a feeder, some of which can be farmers with very different behaviours. The sudden changes in electricity needs, when a farmer starts irrigating, are greatly affecting the distribution compared to having many users requiring small loads but in a more even manner.

Finally, one of the interest in using agent-based modelling for the smart grid is the capability of the agents to be autonomous, to have explicit goals that drive their behaviour and to learn from past experiences [6]. This is an important quality as its components are to automatically adapt to the changes in the network and respond to them so that electricity is delivered in an efficient manner. Rule-based or evolutionary algorithms, for example, can be trialled in the simulation tool before being implemented on the grid.

2) Designing the Software in a Modular Manner: Studying the smart grid requires building a system that can be rather large in terms of the number of components to be modelled and the representation of their interactions. It also necessitates many different types of analyses depending on which aspect of the smart grid a user is interested in.

Consequently, building software for the smart grid will lead to large software systems which will become more and more complex as they are being built. One way of avoiding complex and difficult to extend and maintain software, is to build them in a modular manner. "Modularity involves breaking a large system into separate physical entities that ultimately makes the system easier to understand. By understanding the behaviours contained within a module and the dependencies that exist between modules, it's easier to identify and assess the ramification of change" [7]. That way, not only is it easier to build the system, it is also easier to modify parts of the code when more is learnt about smart grids, as they develop.

Also, from Table 1, having different databases holding the information was identified as a challenge to modelling the grid. Being able to handle all these data types is made easier by having a modular approach, where each element of the software system can use one or many data formats, and common interfaces are used to hide the differences between alternative suppliers of similar data.

III. RELATED WORK

Modularity is not a new concept. Parnas in [8] describes it as information hiding, where separation of concerns [9] are respected. Many programming languages have been developed with this concept in mind, with object-oriented programming one of them. While modularity can be seen at the object level, such as with the object-oriented paradigm, the modularity can happen at a higher level of granularity. For example, component-based software engineering [10] is an example of a modular implementation at the software level. The work presented in this context considers modularity at the software level, i.e. at the component level. However, it also uses modularity at a fine level through the use of agents that are defined in an agent-based model. Unlike EPOCHS [11], our agents are the finest level of granularity of the system, and contain the information describing their behavior through their implementation; an EPOCHS ‘agent would be what we call a module. More detail on modularity in the MODAM context is given below.

IV. THE FRAMEWORK

Having demonstrated the interest of implementing an agent-based model in a modular manner for the simulation of a smart grid, this section describes how such a framework can be built, and what software engineering solutions have been used for this. An example of an implementation is given here to illustrate the principles described; it describes a network containing solar panels. The example describes the different parts of the simulation set up but does not go into details about the whole simulation; more details on the whole simulation can be requested to the authors.

A. Technology - OSGi and Eclipse Plugins

The implementation of the platform was done in Java using the Eclipse Platform [2] . The modularity of the platform was achieved through the use of Eclipse plugins which can be defined as OSGi bundles. OSGi is a specification as defined by the OSGi Alliance (formerly Open Services Gateway Initiative) [1], and OSGI bundles are defined as the unit of modularisation [12]. A bundle is a self-contained unit which explicitly defines its dependencies on other modules and services, as well as its external API.

The notion of modularity is not specific to any technology, and can be achieved using principles in standard Java, for example [13]. However, OSGi offers a higher value solution to the problem of encapsulation, and eases the modularity of a software: "This is where a module framework, such as OSGi, shines because it allows you to carefully encapsulate implementation details within a module through its explicit import package and export package manifest headers" [7], chapter 3. Consequently, the programmer can effectively control the provided API and the dependencies of their plugins. The modules and services can also be dynamically activated, de-activated, updated and de-installed, which makes their use very flexible, especially since it is possible to change the configuration of the system at runtime. For example, it is possible to download a module from a website and add it to a model without recompiling or recoding any module.

B. MODAM Framework

The MODAM (MODular Agent Model) framework is the backbone structure for a modular agent-based model implementation. It is currently only used in the context of the smart grid; however, it is not limited to it and is applicable to other domains such as the water or transport areas. Consequently, some definitions below are quite generic, while examples are always referring to the smart grid.

1) Breakdown of the Software into Reusable Modules: A module in the MODAM framework is defined as:

Module = Name + Assets + Agents + Data

A module is here an Eclipse plugin. As described above, OSGi was the chosen technology, and the framework is implemented using Eclipse plugins as the development unit. Plugins are the smallest deployable and installable software components of Eclipse, and they can define extension points, which allow other plugins to reference and use them.

Figure 1 below shows an example of 2 modules, along with the definition of the extensions from the framework plugin. This example will be used all along the paper to support the description of the different components making MODAM.

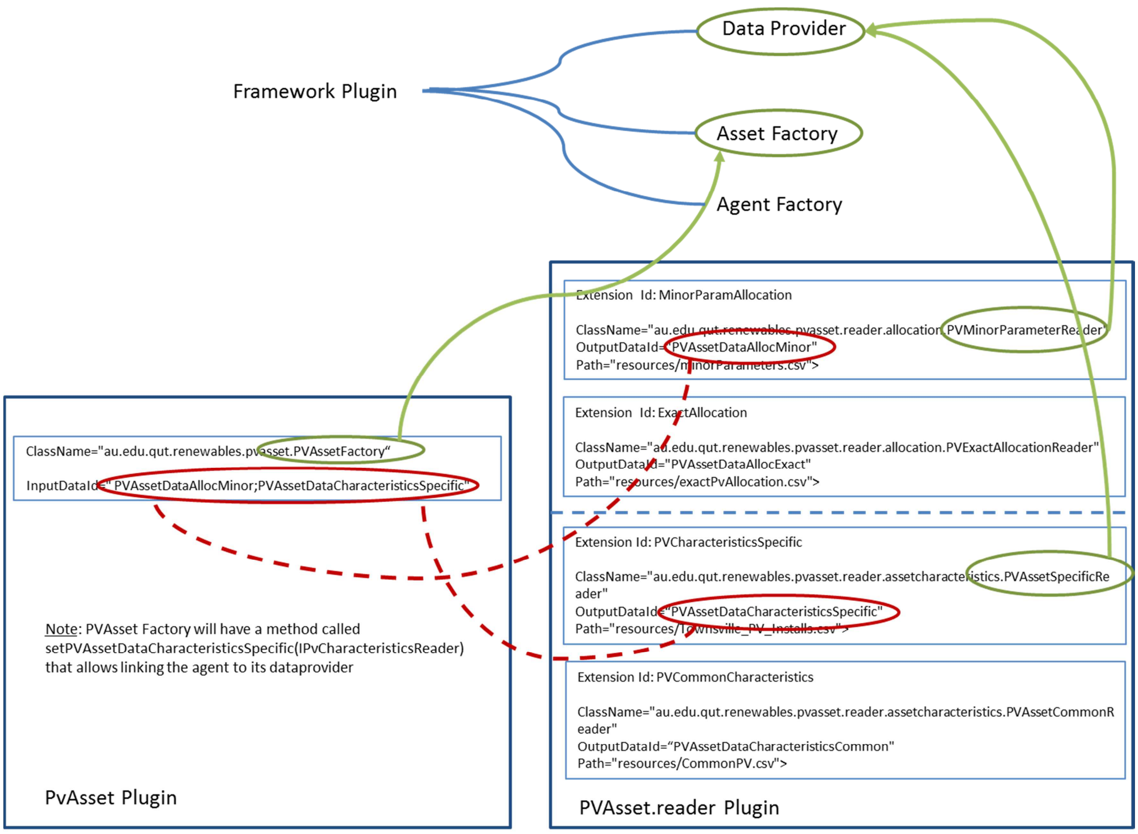


Figure 1– Example of definition of 2 Plugins, and their relationship to one another.

PvAsset Plugin contains an extension point for the asset factory, and PvAsset.reader plugin contains many extension points for the data providers. These are shown with the green arrows, showing an implementation of the interfaces defined in the MODAM framework. These two plugins are linked through their InputDataId and OutputDataId – shown in the red dashed lines. Many extension points can be defined in the plugin.xml, and only used when needed through their linking.

We can see here that the framework plugin defines 3 extensions: Data Provider, Asset Factory and Agent Factory. These are used in any plugin that extends the framework through their plugin.xml file. In Figure 1, PvAsset plugin has a class that extends the Asset Factory, and PvAsset.reader one that extends the Data

Provider; the two modules are also linked through the use of data values. This way, each module satisfies the definition above.

One important feature of this framework is the use of the factory pattern. Assets and agents in a module are not created manually and individually; they are created using factories (implementing the interfaces Asset Factory and Agent Factory). This process is done in an automated manner, using information held by the data providers (more details on this below). This way of creating the agents differ from the classic agent-based model implementations such as Repast [14] or MASON [15] which require the modeller to define agents individually. Here, the agent factory will create all the agents it finds, from the assets already created, not just some specified. Also, binding the module to its data is a requirement of this framework, which is done through the data providers (more detail on this below).

2) Defining the Interfaces from the Modularity Requirements – Extension Points: modularity is used for implementing the model as well as for populating it to build the simulations.

a) Modularity within the Agent-Based Model - Separation of Asset and Agents: One of the first steps towards modularity was the identification of 2 main types of entities in the definition of the model: assets and agents. The assets are the entities that describe the network topology (information about their characteristics and their physical connections) and the agents describe their behaviour. In a classic way, an agent would normally contain both the characteristics and the behaviour in one single object. Here a clear distinction has been made between the assets and the agents, giving more flexibility to the model. For example, the behaviour can be defined as the consequence of a given policy. Having many policies to be tested, these can be implemented in the behaviour and easily tried by assigning them to an asset without needing to modify one or more of the methods of the object.

Also, using this approach, an asset can be assigned 1 or more agents to describe its behaviour, allowing an asset to remain the same even if its user changes its behaviour during its lifetime. An example of this would be of a premise asset that would see a change of tenants and consequently of electricity usage, while still maintaining the same characteristics in terms of insulation when calculating the heating and cooling needs of the building envelope.

b) Modularity when Populating the Model – using Different Data Readers: in order to answer the challenge of dealing with different databases that hold the information, two approaches at least can be taken. One is to implement one type of readers in the software, requiring data manipulation before importing the file. Another is to have a different reader for each of the databases, so that file types are dealt with individually to populate the model. The second option was chosen in the MODAM implementation as it offers more flexibility; the first one is still possible however. The extension point for Data Provider gives the flexibility to having different data formats as many implementations of a data provider can be provided. The object needing access to the data can then call the interface without needing to know anything about the data format. If required, the data provider implementation can be switched over without impacting the rest of the code.

c) Connecting Modules Together through Interfaces: In Eclipse, plugins are connected through extension points and extensions. The extension points defined in MODAM are given in Figure 2, following the 3 requirements described above (separation of asset and agent, and different data readers). These extension points define interfaces in the code, allowing the different modules to be connected to one another, ensuring the functioning of the modular platform. The definitions of these extensions are given in Table 2, where the attributes are described, and the interfaces are given in the ClassName attribute of the extensions. These interfaces differentiate this approach from the traditional tools used for agent-based modelling, such as MASON and Repast [16], which are non-modular and combine the data and behaviour aspects of agents.

It has to be noted that additional extensions can be defined if required, and this can be done in the users' modules too. This ensures the extensibility of the MODAM framework, allowing users to build on the existing code.

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
  <extension-point id="dataprovider" name="Data Provider" schema="schema/datareader.exsd"/>
  <extension-point id="agentfactory" name="Agent Factory" schema="schema/agentfactory.exsd"/>
  <extension-point id="assetfactory" name="Asset Factory" schema="schema/assetfactory.exsd"/>
</plugin>
```

Figure 2 – Extension points for the MODAM framework – extract from the plugin.xml file.

Table 2 –MODAM extension definitions

Asset Factory	Agent Factory	Data Provider
extension Sequence (1 - *) AssetFactoryClass point id name AssetFactoryClass ClassName InputDataId Predecessors	extension Sequence (1 - *) AgentFactoryClass point id name AgentFactoryClass ClassName InputDataId	extension Sequence point id name DataReader ClassName OutputDataId DefaultPath

3) Connecting the Modules Together When Setting Up a Simulation: Having defined the different elements of the code in the different plugins, these need to be connected to one another. This is done by the module manager that can be seen as the central point of code. Figure 3 shows the lifecycle of the simulation tool. Two parts are distinguished here; the one setting up the simulation (Module Manager) and the one running it (ABM state). The role of the Module Manager is to find all the plugins that are available in the registry, and enable those that have been chosen by the user. From these enabled plugins, the required plugins that are missing are found and added to the simulation.

The module manager can then call the asset factories and the agent factories that have been passed through the extension points of the enabled plugins. Methods are then called on these factories, using reflection, by just knowing the type of interface they implement – these have been passed in through the extension point definition.

Once all the modules have been enabled, the simulation can be started (ABM state part of the lifecycle). The simulation can be started and stopped at any time; the simulation is running when the step() method of the agents is called upon.

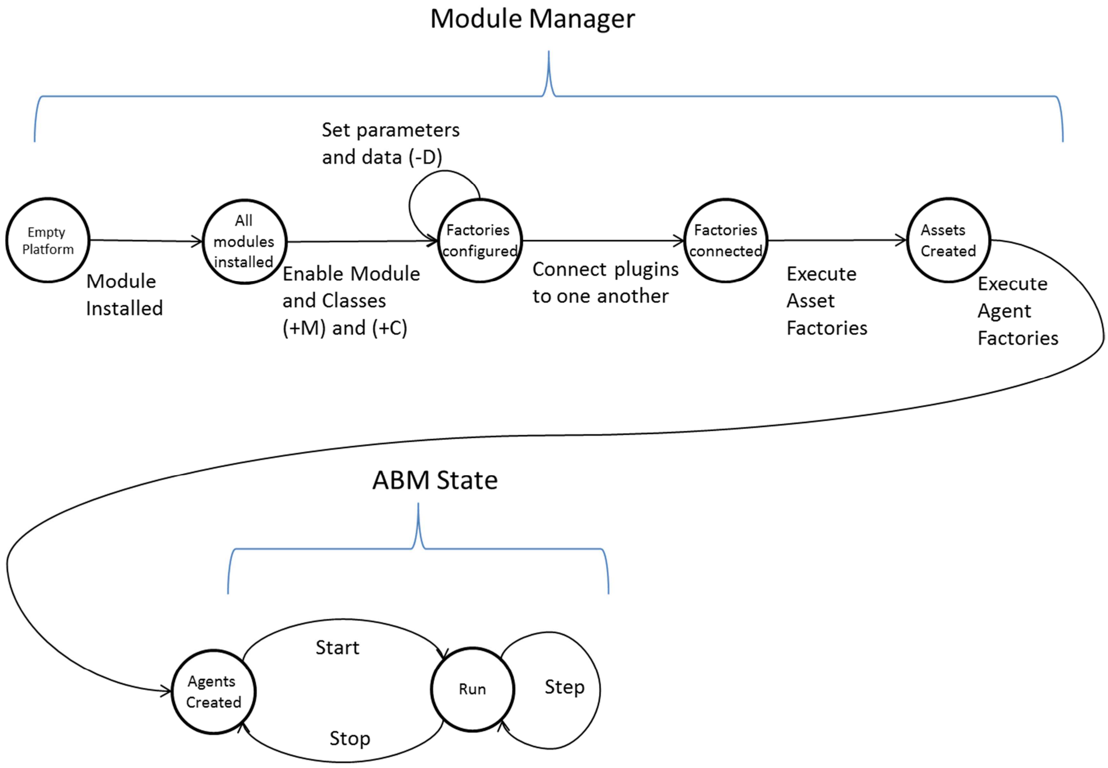


Figure 3 – Illustration of the lifecycle of the simulation tool.

2 phases can be distinguished here: the set-up of the simulation (Module Manager part) and its running phase (ABM state). The Module Manager is responsible for finding all the modules, connecting them together and creating the assets and agents through their factories; data is used to populate them. The simulation can then be started and stopped as required.

Through these extension definitions, the plugins can be linked not only by extending the interfaces, but through the data values. These data values are defined in the plugin.xml file under the InputDataId and the OutputDataId attributes, and need to have the same value to be linked to one another.

For example, in Figure 1, PvAssetFactory will use the data provider PVAssetSpecificReader that has been linked to it through the PVAssetDataCharacteristicsSpecific value in the PVAsset.reader plugin; and the data provider PVMinorParameterReader linked through PVAssetDataAllocMinor. These links are shown in Figure 1 through the dashed lines. These 2 extension points are used in the code to describe the types of solar panels that need to be created

(PVAssetDataCharacteristicsSpecific), and to which asset they are associated with, i.e. premise or substation number (PVAssetDataAllocMinor).

It can be seen as well in Figure 1 that there are 4 extension points for the PvAsset.Reader plugin, but only 2 are used. This demonstrates the flexibility in the model implementation, as different combinations can be chosen to create the solar panels, e.g; it is therefore possible to change the behaviour of the software in an easy way.

While these plugins are linked through the plugin.xml file, their use in the code is automated through the use of reflection in the Module Manager. Consequently, the user does not need to modify the Module Manager, but simply define these 2 attributes with the same value in the plugins and implement a method named “set + ValueOf(OutputDataId)” in the factory class. For example, according to Figure 1, the class PVassetFactory then requires 2 methods that will be:

- setPVAssetDataAllocMinor (IDataProvider)
- setPVAssetDataCharacteristicsSpecific (IDataProvider)

Finally, an InputDataId can have many values as shown in Figure 1, which are separated by semi-colons.

4) Cross-Connections amongst Plugins: As in many software systems, parameters can be set and used at different stages of the simulation. MODAM considers two types of parameters: global and plugin specific. As its name indicates, the plugin specific parameters will be held at the plugin level and cannot be accessed from other plugins. An example of this would be whether the power flow analysis is done using simple or complex power in the power flow plugin. The global parameters however, are defined in the module manager and can be tracked all along the simulation from any plugin. Such parameters are the start time, end time, and random seed.

An asset can be used by different modules by having an agent in each of these modules. For each agent type, different attributes of the asset would be used that can be defined at runtime through the use of channels. Channels are defined here as a set of attributes defined at runtime for an agent. Depending on the value of the channel parameter, the behaviour of the agent is different, through its connection to a different demand data type (residential, commercial data…) or different logic (simple and complex power). While the choice of the parameter is defined at runtime and within a plugin, all the available channels are defined globally and held in the MODAM framework.

The data structure chosen for the channels is a map object where a given parameter will be assigned a value which can be accessed anywhere. This map is located at the ABMState level. The reason for doing this was so that someone who needs to use the MODAM framework in the future will have the mechanisms to use global variables in that manner.

5) Setting up the Simulation – Command Line Arguments: In order to run a simulation, modules relevant to the specific analysis can be loaded. For this, a command line reader was created; an example of it is given in Table 3.

Table 3 – Example of 2 Simulations set up, using command line and configuration.

Command Line example for network simulation	Reuse of an existing configuration file plus additional commands
+M= assetreader +C=assetreader.NetworkReader +C=assetreader.LocationReader +M=demandreader +C=demandreader.historical.HistoricalDemandReader +C=demandreader.billing.BillingDataReader +M=assetnetwork +C=assetnetwork.ergon.NetworkAssetFactory +C=assetnetwork.agent.NetworkAgentFactory -from=2010-01-01 -to=2010-01-08 -output=tempOutDir	-config=network.xml +M=pvasset +C=pvasset.PVAssetFactory +M=pvagent +C=pvagent.WeatherPVAgentFactory +M=pvasset.reader +C=pvasset.reader.allocation.PVMinorParameterReader +C=pvasset.reader.assetchar.PVAssetCommonReader +M=weatherreader +C=weatherreader.CloudDataReader +C=weatherreader.TemperatureDataReader -output=tempOutDir

First, the modules and their classes required to set up the simulation are given, using "+M" and "+C" respectively followed by the names of the required modules and classes. Specifying the classes is not always required, for example when only one type of factories is available in the module. In that case, only the "+M" command will be called. However, if there are different factories in one module for example to describe the network, e.g. network data and SWER data, a distinction can be made as to which needs to be called. Each of the specified classes can also be parameterised using the "-D" command followed by the parameter value. This is then called by the class as an argument and using reflection on the parameter name. For example, ‘-D = AllocationMethod = “R”’ will be used in the specified class with the method setAllocationMethod (String R). Finally, other parameters for the simulation run can be passed. These are the start and end times of the simulation, called using “-from” and “-to”. And a folder that will contain the output of the simulation can be specified using “-output”.

While many modules can be created, not all of them need to be loaded, only those required for a given analysis type. However, as the model grows, many modules might be required to be loaded as they will ensure that the whole of the system is taken into account. This might lead to a very long command line. To prevent this, and also build on previous simulation runs, it is possible to use a configuration file that has been saved in a previous simulation. An example of this is also given in Table 3, which calls “-config” with the name of the file, and adds the new modules that are required for this simulation.

To summarise, Table 3 shows the command line for 2 simulation runs. The first one is to run the demand on a network and uses information from 3 plugins which create the assets and the agents using the data provided by the readers. The simulation is to be run for 1 week, from the 01/01/2010 until the 08/01/2010, and the output of the simulation will be saved in the tempOutDir directory. The second simulation builds on this one (calling the network.xml file), and 4 additional modules are loaded. These 4 modules are for the modelling of the solar panels.

V. APPLICATION OF MODAM

Using the approach described above, many simulations have been performed, investigating different parts of the system. For example, in addition to the 3 phase network that is mostly found in cities, simulations on a SWER network were performed to study the load variations on a rural network in central Queensland. The voltage drops seen by each customer as the load varies were calculated using a load flow analysis. A battery plugin was added where battery assets could be placed on the network to support voltage drop at places under stress. By adding other plugins that describe the batteries behaviour, different control algorithms could be tried to identify the ones that would be most helpful to the network.

When assessing the impact of renewables on the grid, and more particularly describing the behaviour of solar panels, many different approaches can be taken.

One is to use historical data for given solar panels and reuse them in future years, expecting simular weather output. Another one is to simulate the PV output using weather information as well as the usual physical equations, and predict the PV output taking into account the passage of clouds – details on this implementation can be found in [17]. These 2 approaches have been implemented in 2 distinct plugins and can be selected indifferently depending on the needs of the user.

With time, it is expected that many more plugins will be added so that the behaviour of the system can be captured in its entirety. One of the near future tasks is to explore different types of demand-side management (DSM) and their uptake level on the grid. Each of these DSM options is expected to be implemented in separate plugins, and called at setup of the scenario depending on the type of assessment required. Most of the power grid scenarios are handled by adding new modules and/or extending the existing modules to have flags and parameters to give more control over their behaviour.

VI. CONCLUSION

Smart grids can be modelled using agent-based modelling in a modular manner which is an efficient manner of building software. Taking such an approach allows building on previous work, as the simulation environment grows and more data becomes available. This paper demonstrated that such an approach is possible through the illustration of the implementation of the functionalities on a network with solar panels. The code for the MODAM framework which is open-source can be used for the implementation of user functionalities of the smart grid as more data become available. Examples of functionalities of the smart grid, such as feedback loops have not been shown here, because the aim of this paper was rather to set the architecture for a modular approach to agent-based modelling in the view of simulating the smart grid rather than the different algorithms that populate the software. More details on the implementations of the functionalities will be given in a later paper. This paper showed that current software engineering techniques can be useful in developing solid software for the smart grid.

REFERENCES

[1] OSGI Alliance. (2013, 01/02/2013). *OSGI Alliance*. Available: http://www.osgi.org/Main/HomePage

[2] The Eclipse Foundation. (2012, 27/02/2012). *About the Eclipse Foundation*. Available: http://www.eclipse.org/org/

[3] The National Institute of Standards and Technology (NIST). (2012, 06/02/2013). *Smart Grid: a Beginner's Guide* Available: http://www.nist.gov/smartgrid/beginnersguide.cfm

[4] F. A. Boulaire, M. Utting, R. Drogemuller, G. Ledwich, and I. Ziari, "A Hybrid Simulation Framework to Assess the Impact of Renewable Generators on a Distribution Network," in *2012 Winter Simulation Conference*, Berlin, Germany, 2012, pp. 1-12.

[5] L. A. House-Peters and H. Chang, "Urban Water Demand Modeling: Review of Concepts, Methods, and Organizing Principles," *Water Resources Research,* vol. 47, p. W05401, 2011.

[6] C. M. Macal and M. J. North, "Agent-Based Modeling And Simulation," in *2005 Winter Simulation Conference*, 2005.

[7] K. Knoernschild, *Java Application Architecture: Modularity Patterns with Examples Using OSGi*: Prentice Hall, 2012.

[8] D. L. Parnas, "On the criteria to be used in decomposing systems into modules," *Communications of the ACM,* vol. 15, pp. 1053-1058, 1972.

[9] E. W. Dijkstra, "EWD 447: On the role of scientific thought," *Selected Writings on Computing: A Personal Perspective,* pp. 60-66, 1982.

[10] C. Szyperski, *Component software: beyond object-oriented programming*. New York: ACM Press, 1997.

[11] K. Hopkinson, W. Xiaoru, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "EPOCHS: a Platform for Agent-Based Electric Power and Communication Simulation Built from Commercial Off-the-shelf Components," vol. 21, pp. 548-558, 2006.

[12] L. Vogel. (2012, 24/09/2012). *OSGi Modularity - Tutorial*. Available: http://www.vogella.com/articles/OSGi/article.html

[13] K. Knoernschild. (2012, 04/12/2012). Patterns of Modular Architecture. *DZone Refcardz,* 7. Available: www.dzone.com

[14] Argonne National Laboratory. (2011, 25/09/2011). *Repast Simphony*. Available: http://repast.sourceforge.net/repast_simphony.html

[15] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan, "MASON: A Multi-Agent Simulation Environment," *Simulation: Transactions of the society for Modeling and Simulation International,* vol. 82, pp. 517-527, 2005.

[16] C. Nikolai and G. Madey, "Tools of the Trade: A Survey of Various Agent Based Modeling Platforms," *Journal of Artificial Societies and Social Simulation,* vol. 12, p. 2, 2009.

[17] F. A. Boulaire, M. Utting, R. Drogemuller, A. Abeygunawardana, G. Ledwich, and J. Bell, "Planning for the Impact of Distributed Solar Energy on the Grid," presented at the Solar 2012 Conference, Swinburne University of Technology, Melbourne, 2012.